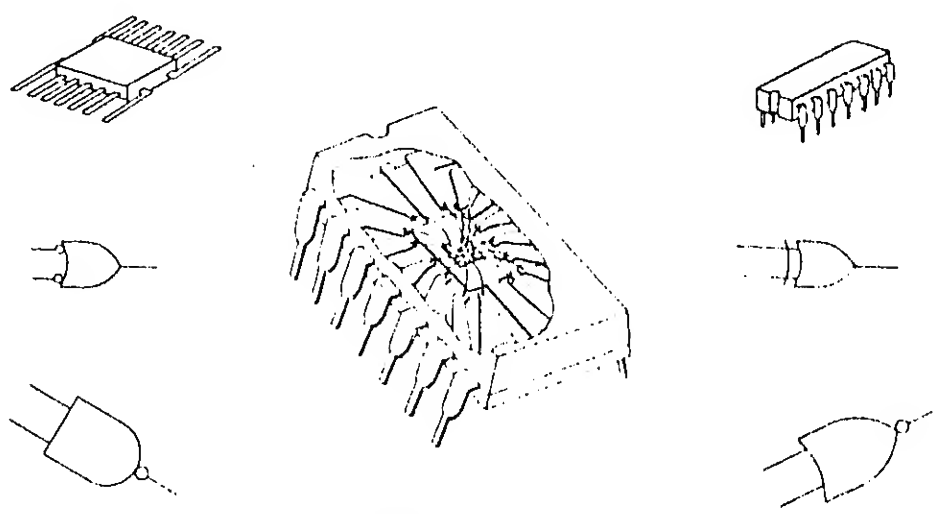
STUDENT'S GUIDE
FOR
ADVANCED FIRST-TERM AVIONICS COURSE

CLASS A1

C-100-2010



UNIT V

PREPARED BY

NAVAL AIR TECHNICAL TRAINING CENTER
NAVAL AIR STATION MEMPHIS
MILLINGTON, TENNESSEE

PREPARED FOR

NAVAL TECHNICAL TRAINING COMMAND

CNTT-M1635 Rev.12/84

JANUARY 1983

FOREWORD

The purpose of this Student's Guide is to assist you through Digital Computer Theory and Troubleshooting, Unit 5, of the Advanced First-Term Avionics Course. The proper use of this guide will not only sharpen your theoretical troubleshooting skills, but will also aid you in the lab with actual hands-on practical work, troubleshooting the COM-TRAN Ten digital computer down to a defective component.

The table of contents lists the page numbers for safety notices, Assignment Sheets, Information Sheets, Job Sheets, and Notetaking Sheets that will further enhance your abilities and skills as electronics technicians.

SAFETY NOTICE

As a Navy electronics technician, you will be required to perform safe and efficient maintenance on various types of electronic equipment. Not only your life, but the lives of many others will depend on your being safety conscious at all times. It is the responsibility of all Navy and Marine Corps personnel to prevent accidents. This can be done if everyone develops conscientious safety habits and observes all precautions when performing maintenance of any type.

HOW TO USE THIS STUDENT'S GUIDE

This Student's Guide has been prepared for you to use while you are attending the Advanced Term Avionics (A1) Course. Ample space has been provided for taking notes on the required lesson information. Remember, when you are in class, the information being provided by your instructor is information you will need in performing your Navy job.

This volume contains the following:

1. Assignment Sheets for most topics, to direct your study efforts.

2. Information Sheets for some lessons, to provide additionl material for more difficult lesson topics.

3. Notetaking Sheets, containing lesson topic outlines will illustrations and ample space for personal notetaking.

4. Job Sheet, to direct you in your first laboratory assignment.

GOOD LUCK! Learn all you can!!

UNIT V CLASS SCHEDULE

Unit V is two weeks long, and starts in the middle of the fifth day of the eighth week. The periods run from 317 to 396, with the last period finishing half-way through the last day of the teneth week.

The schedule is as follows:

TOPIC NO.	TYPE	PERIOD	TOPIC
EIGHTH WEEK			
Fifth Day			
4.17	Class	313	Unit/Module Test: Criterion Test/Written Examination
		314	
		315	
		316	
5.1	Class	317	Introduction to Digital Computers
		318	
5.2	Class	319	Mathematics of Digital Computers
		320	
NINTH WEEK			
First Day			
5.2	Class	321	Mathematics of Digital Computers
		322	
		323	
		324	
5.3	Class	325	Basic Logic Gate Interpretation
		326	
		327	
		328	
Second Day			
5.3	Class	329	Basic Logic Gate Interpretation
		330	
5.4	Class	331	Introduction to the COM-TRAN Ten
		332	Computer and Organization
		333	
5.5	Class	334	COM-TRAN Ten Logic and Data Flow
		335	
		336	

TOPIC NO.	TYPE	PERIOD	TOPIC
Second Day			
5.8	Lab	369 370 371 372	COM-TRAN Ten Data Flow Analysis
5.9	Lab	373 374 375 376	COM-TRAN Ten Fault Isolation
Third Day			
5.9	Lab	377 378 379 380 381 382 383 384	COM-TRAN Ten Fault Isolation
Fourth Day			
5.9	Lab	385 386 387 388	COM-TRAN Ten Fault Isolation
	Lab	389 390 391 392	Unit/Module Test: Criterion Test/Performance Test
Fifth Day			
	Class	393	Unit/Module Test: Criterion Test/Written Test
		394	Within-Course Comprehensive Test 1
		395	Written Examination and Review
		396	
6.1	Class	397	Introduction to Airborne Radar
		398 399	
		400	(Demo)

UNIT V HOMEWORK SCHEDULE

All of the Assignment Sheets listed below shall be turned in when due. Each Assignment Sheet will be checked by an instructor for completeness and correctness. Failure to turn in an Assignment Sheet could result in disciplinary action.

Assignment Sheet	Period Due
5.1.1A	321
5.2.1A	329
5.3.1A	337
5.4.1A	337
5.5.1A	337
5.6.1A	345
5.7.1A	369

UNIT V LEARNING OBJECTIVES

TERMINAL OBJECTIVE

11.0 ISOLATE an instructor-induced malfunction (under limited supervision) in an avionics, general purpose digital computer training device to a weapons replaceable assembly, a shop replaceable assembly, a stage, and a component and RECORD results on job sheets. Test equipment will be provided. Performance must be accomplished in accordance with COM-TRAN Ten Technical Operations Manual M104, Vol. I. All general and personnel safety precautions must be observed, in accordance with OPNAVINST 5101.2 (series).

ENABLING OBJECTIVES

11.1 EXTRACT troubleshooting and performance data from given block and logic diagrams of a general purpose digital computer training device. All circuit performance and operating characteristics will be documented on job sheets in accordance with specifications contained in COM-TRAN Ten Technical Operations Manual M104, Vol. I.

11.2 PERFORM visual inspections on an avionics general purpose digital computer training device for physical defects, security, integrity, and proper installation and RECORD results on a job worksheet. Performance must be accomplished in accordance with procedures outlined in COM-TRAN Ten Technical Operations Manual M104, Vol. I.

11.3 PERFORM operational and minimum performance chceks (under limited supervision) on an avionics general purpose digitl computer training device and RECORD results on job data sheets. Necessary test equipment will be provided. Performance must be accomplished in accordance with COM-TRAN Ten Operations Manual M104, Vol. I. All safety precautions must be observed in accordance with OPNAVINST 5101.2 (series).

11.4 ISOLATE an instructor-induced malfunction (under limited supervision) on an avionics general purpose digital computer training device to a weapons replaceable assembly, a shop replaceable assembly, a stage, and a component and RECORD results on job sheets. Test equipment will be provided. Performance must be accomplished in accordance with the COM-TRAN Ten Technical Operations Manual, Vol. I. All safety precautions must be observed in accordance with OPNAVINST 5101.2 (series).

11.5 DOCUMENT, on the VIDS/MAF, all necessary corrective actions required in a given maintenance situation to restore an avionics general purpose digital computer training device to an operational condition. Documentation must include the ordering and receipt of parts. All documentation must be legible and in accordance with OPNAVINST 4790.2 (series).

The other branch of the computer family stems from the graphic solution of the straighedge and compass problems achieved by ancient surveyors. Analogies were assumed between the boundaries of property and on the lines drawn on paper by the surveyor. Devices that relay on the analogous relationships that exist between the physical quantities associated with a problem under study are called analog computers. Analog solutions are obtained by measurement of some continually varying quantity, such as the fuel in a tank or the value of a modulated voltage. Such devices are referred to as continuous computers.

Accuracy of computers

The fundamental difference between analog and digital devices is that digital computers deal with discrete quantities while analog computers deal with continuous physical variables. The accuracy of an analog computer is restricted to the accuracy with which physical quantities can be sensed and displayed. This in turn is related to the quality of the components used in construction the computer; for example, the tolerance of electrical resistors or mechanical shafts, which would control the quality of the output representation.

The concept of analog computation

The most significant feature that distinguishes analog computation from numerical computation is continuity; analog computation implies continuous computation; that is, the variables involved in a calculation appear as continuous or smooth functions. As an example, consider the common clock as a computing device. Its function is to give an indication of elapsed time. It does this either by simple addition or by integration. A clock that keeps track of elapsed time by adding a discrete angular displacement to the hands each time a pendulum or balance wheel reverses direction or a timing pulse is received from a central control unit is a digital-computing device. (Almost all watches fall into this category.) The output of such a clock, an indication of elapsed time, changes in discrete steps. The size of the discrete steps varies according to the mechanism producing them. Some grandfather clocks advance the hands only once every minute; high-frequency pulse-controlled clocks change their output at such a rapid rate that to the observer the output seems continuous. A clock that changes the position of the hand continuously with time, such as an electric clock, is an analog-computing device. It integrates, with respect to time, the angular velocity of the motor shaft and obtains a smooth, continuously changing angular velocity of the motor shaft and obtains a smooth, continuously changing angular displacement of the hands. Thus one type of clock computes elapsed time digitally by counting the number of oscillations of a balance wheel or pendulum; the other

type computes in analog fashion by continuous integration. A little throught will convince the technician that the latter is a rather natural process. Very few phenomena in the physical world change in discrete steps. The world is made up of quantities or variables that are continuous functions of time. These change smoothly and continuously.

Development of digital computers

In the fields of science and industry, there has been a growing need for high-speed computing devices. Long, tedius calculations that take months to perform have impeded, to a considerable extents, the progress or scientific research and development. Time-consuming computations that might take a human being years to do can be accomplished in a matter of minutes with modern high-speed computers. It is known that computers, when supplied with pertinent data, can predict political outcomes, guide missiles, determine production improvements, play games of chess and checkers, write music, and solve problems at fantastic speeds. How these things are accomplished will be increasing interest to the avionics ratings.

The purpose of this course is to remove much of the mystery surrounding digital computers by delving into their methods of operation and by familiarizing the student with the language of the science.

Functions of digital computers

Needless to say, digital computers, like most recent discoveries, are not new. They have been used as long as man has been holding up fingers to designate quantity. Only the methods have changed. Down through the centuries man has used fingers, adding machines, calculators, and now, computers. The methods used by each are different, but the final outcome is the same. The answers are based on logical calculations.

Dispel any thoughts that a computer can think--it cannot. Digital computers, when manufactured, are as devoid of knowledge as a door knob. Only after some information is placed into them do any calculating. A computer does not know that 2 plus 2 equals 4; however, it definitely has the ability to make that calculation. The ability is built into the computer by various arrangements of relatively simple circuits. When a computer performs an operation, it does so by comparing quantities representing data that have been placed into it. Its output consists of coded numbers that must be read an automatically decoded into some form of understandable language.

Data representation in digital computers

The word "numbers" as applied to computers is misleading--coded numbers would be more descriptive of the machine language. Acutally numbers that represent information are merely a series of pulses and the absence of pulses, cleverly encoded to represent various letters, numbers items, operations, directions, or whatever else is desired.

Anything can be represented by a code of a sort; for example, letters of the alphabet, numbers, towns, distances. Coded messages are nothing more than letters and numbers represented in other forms, or terms. In coded messages, the letter "A" may mean originator, the number "6" may mean the letter "H," etc. In digital computers, some code bases on binary numbering is used. By employing binary 1's and 0's, represented by the presence and absence of electrical pulses, various things, such as letters and numbers, are represented. The reason for this method of coding will become apparent as you progress through this course.

Description of digital computer units

Digital computers contain blocks of circuits and machinery (see figure 1) that can be broadly broken down into:

1. Input unit.

2. Output unit.

3. Memory unit.

4. Arithmetic and logic unit (ALU).

5. Control unit.

The input unit is just what the name implies. It is the unit through which information is fed to the computer. Many methods of feeding information to the computer input unit are used, such as a typewriter keyboard or magnetic tape. They all do the same thing--feed data to the computer so that a logical solution to a problem may be obtained.

The output unit translates the results of the computer's operations into a readable language or into some other form that is readily usable.

The memory unit stores information until it is needed. Most data fed to the computer will be directed to the memory unit. Instructions, intermediate results, and processed data are stored in the memory, as well as raw input data.

The arithmetic and logic unit contains circuits that are relatively simple. It accomplished mathematical calculations and performs logical operations. The arithmetic and logic unit is able to make logical decisions by comparing quantities of data against each other. The solution to a given problem is sent from the arithmetic and logic unit to the memory unit, where it remains until needed for further calculations or until otherwise routed by the control unit.

The control unti decodes instructions and generates timing and control signals, which cause the various units of the computer to function as an integrated system.

Terms and definitions

AUTOMATIC DATA PROCESSING (ADP)--processing performed by automatic means.

ANALOG--pertaining to representation by means of continuously variable physical quantities.

ANALOG COMPUTER--(a) a computer in which analog representation of data is mainly used; (b) a computer that operates on analog data by performing physical processes on these data.

ARITHMETIC UNIT--the unit of a computing system that contains the circuits that perform arithmetic operations.

BIT--a binary digit.

BYTE--a sequence of adjacent binary digits operated on as a unit usually shorter than a computer word. Most modern minicomputers have an 8-bit byte.

CENTRAL PROCESSING UNIT--a unit of a computer that includes the circuits controlling the interpretation and execution of instructions; synonymous with mainframe; abbreviated CPU.

COMPUTER--a data processor that can perform substantial computation, including numerous arithmetic or logic operations, without intervention by a human operator during the run.

COMPUTER GENERATIONS--computers are sometimes classified by the generation of their hardware. Generally they are classified as follows:

a. First generation, vacuum tubes and relays.

b. Second generation, semiconductors discrete devices.

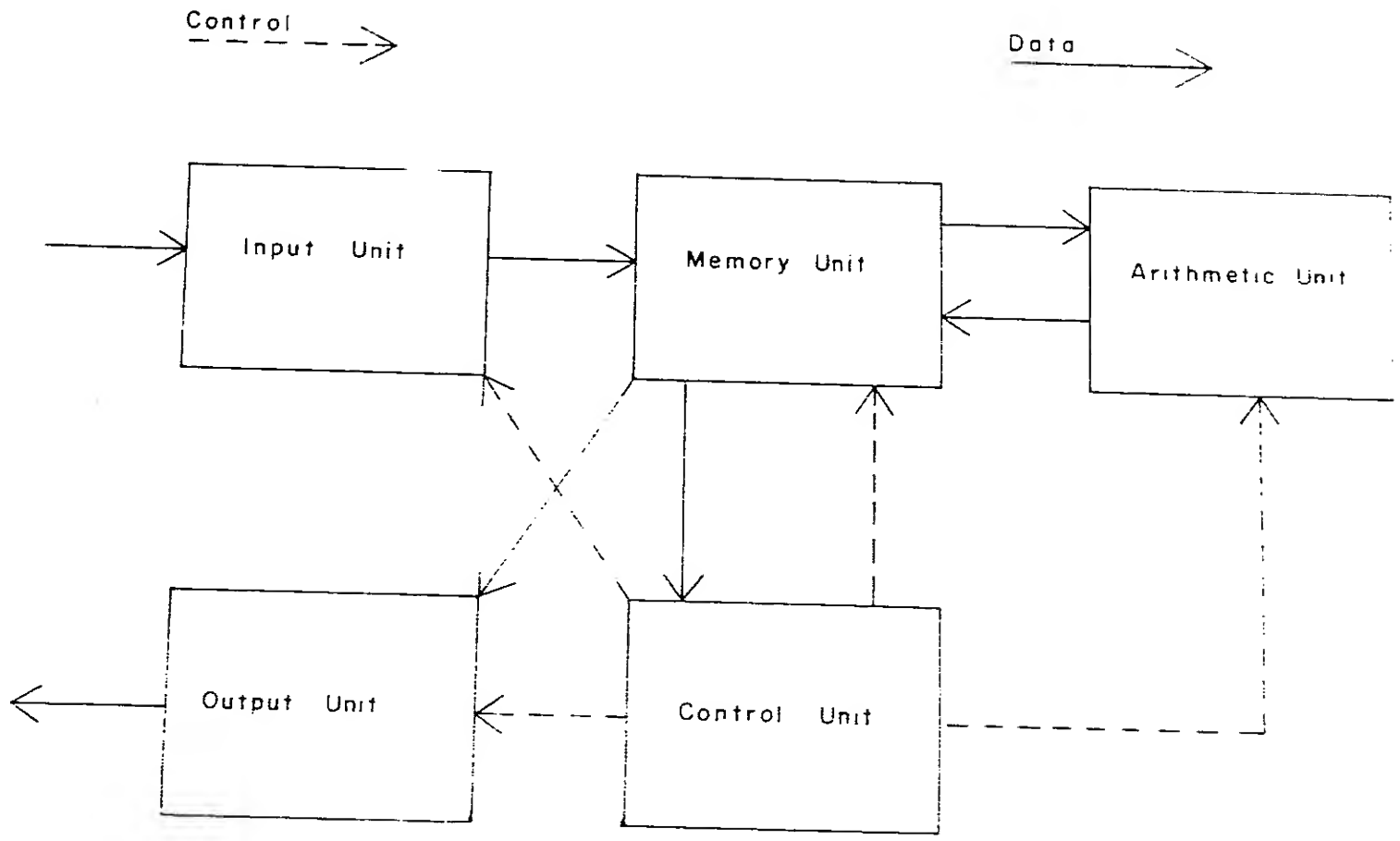


FIGURE 1 - Basic computer block.

c. Third generation, integrated circuits.

d. Fourth generation, large-scale integration.

CONTROL UNIT--a main functional unit of a CPU (central processor unit) responsible for instruction decoding and command and timing generation.

DATA--(a) a representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means; (b) any representation, such as characters or analog quantities, to which meaning is or might be assigned.

DATA PROCESSING--the execution of a systematic sequence of operations performed upon data: synonymous with information processing.

DATA PROCESSOR--a device capable of performing data processing, including desk calculators, punched card machines, and computers.

DIGITAL--pertaining to data in the form of digits.

DIGITAL COMPUTER--(a) a computer in which discrete representation of data is mainly used; (b) a computer that operates on discrete data by performing airthmetic and logic processes on these data.

DISCRETE--pertaining to distinct elements or to representation by means of distinct elements, such as characters.

EXPLICIT ARITHMETIC--adding two numbers to form a total.

FLOWCHART--a graphic representation for the definition, analysis, or solution of a problem, in which symbols are used to represent operations, data, flow, equipment, etc.

GENERAL PURPOSE COMPUTER--a computer that is designated to handle a wide variety of problems. What it can do is mainly limited by its programs.

HARDWARE--physical equipment, as opposed to the computer program or method of use; e.g., mechanical, magnetic, electrical, or electronic devices. Contrast with software.

HYBIRD COMPUTER--a computer for data processing, using both analog representation and discrete representation of data.

IMPLICIT ARITHMETIC--comparing two numbers to determine which is the larger.

INFORMATION--the meaning that a human assigns to data by means of the known conventions used in their representation.

INPUT DEVICES--the device or collective set of devices used for conveying data into another device.

INTERFACE--a shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

MEMORY--(a) pertaining to a device into which data can be entered, held, and can be retrieved at a latter time; (b) loosely, any device that can store data; (c) synonymous with storage, one of the main functional units of a CPU (computer processor unit).

NONVOLATILE MEMORY--a device that retains all the information it contains when power is removed.

OFFLINE--pertaining to equipment or devices not under control of the central processing unit.

ONLINE--(a) pertaining to equipment or devices under control of the central processing unit; (b) pertaining to a user's ability to interact with a computer.

OUTPUT DEVICE--the device or collective set of devices used for conveying data out of another device.

PERIPHERAL EQUIPMENT--in a data processing system, any unit of equipment, distinct from the central processing unit, which may provide the system with outside communication.

PROCESSOR--(a) in hardware, a data processor; (b) in software, a computer program that includes the compiling, assembling, translating, and related functions for a specific programming language.

PROGRAM--a series of instructions or statements, in a form acceptable to a computer, prepared in order to achieve a certain result. A person who is involved in the designing, writing, and testing of programs is called a programmer, and what he does is called programming.

SOFTWARE--a set of computer programs, procedures, and possibly associated documentation concerned with the operation of a data processing system; e.g., compilers, library routines, manuals, and circuit diagrams.

SPECIAL PURPOSE COMPUTER--a computer that is designed to handle a restricted class of problems.

TERMINAL--a point in a system or communication network at which data can either enter or leave.

VOLATILE MEMORY--a device that loses all the information it contains when power is removed.

NOTETAKING SHEET 5.1.1N

INTRODUCTION TO DIGITAL COMPUTERS

REFERENCES:

1. Burroughs Corporation. Digital Computer Principles. N.Y.: McGraw-Hill Book Company, 1969. Chapters 20, 21, 22, and 23.

2. Digital Computer Basics. NAVTRA 10088-B1, 1983.

3. Malvino Albert P., and Donald P. Leach. Digital Principles and Applications. N.Y.: McGraw-Hill Book Company, 1975. Chapter 14.

NOTETAKING OUTLINE

I. Terms and Definitions

II. Classes of Computers

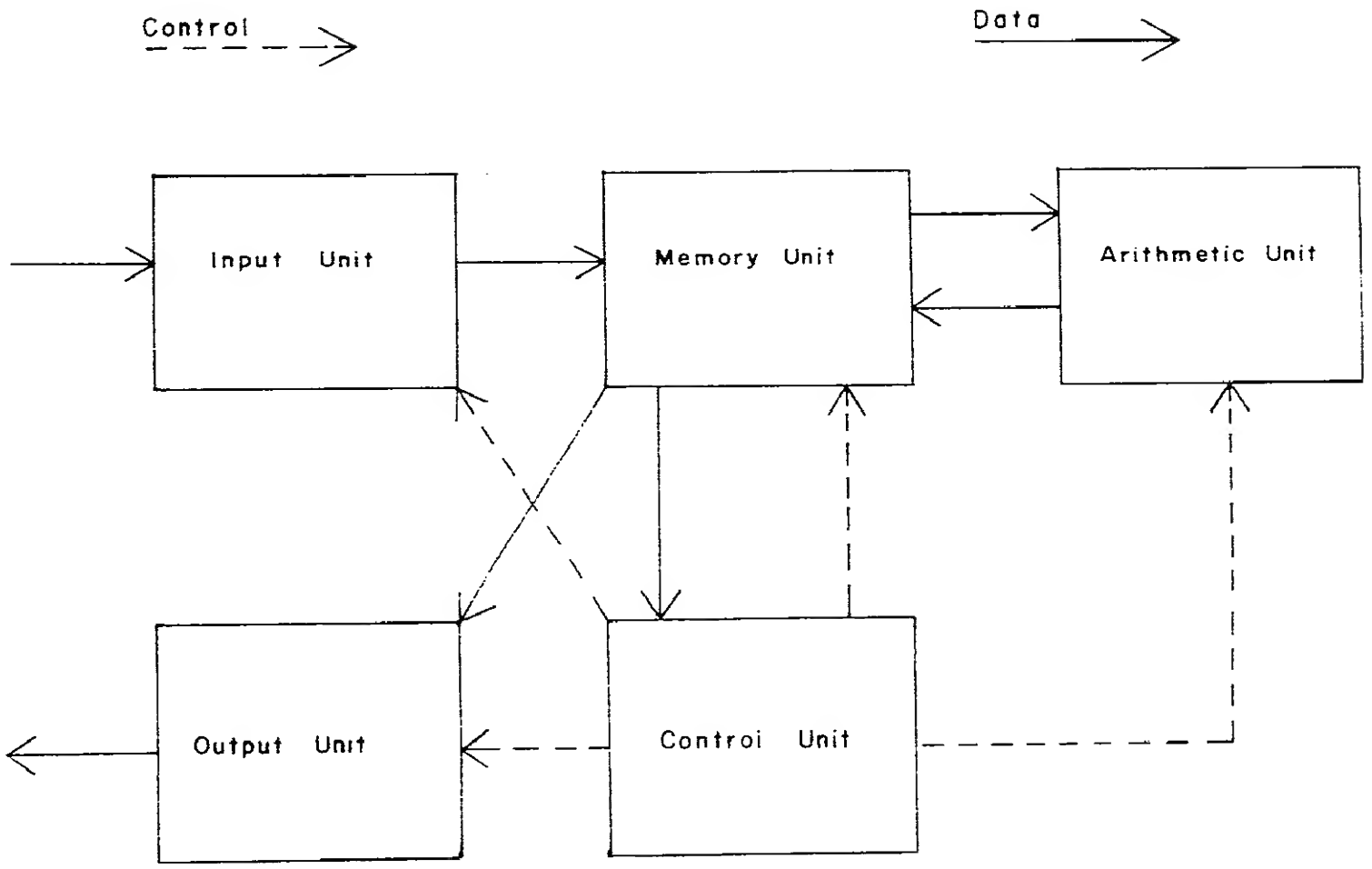


Figure 1 - Basic Computer Block

Types of Digital Computers

Basic Computer Block

INFORMATION SHEET 5.2.1I

MATHEMATICS OF DIGITAL COMPUTERS

INTRODUCTION

Digital computers operate in binary numbers, binary-coded numbers, and binary-related-number systems, such as octal and hexadecimal. Familiarity with the structure and handling of these numbering systems is essential in understanding the arithmetic unit of a digital computer. The purpose of this information sheet is to provide you, the technician, with the basic concepts of computer numbering systems and the computer process of arithmetic.

REFERENCES

1. Digital Computer Basics. NAVTRADEC 10088-B1, 1983.

2. Malvino, Albert P., and Donald P. Leach. Digital Principles and Applications. N.Y.: McGraw-Hill Book Company. Chapter 2, pages 17-34; Chapter 13, pages 365-398.

3. Williams, Gerald. Digital Technology. Chicago: Science Research Association. 1982.

INFORMATION

1. Terms and definitions

   a. Radix (base)--the number of characters used in a numner system.

   b. Radix point--the dividing point between whole numbers and fractions.

   c. Bit--a binary digit--either 0 or 1.

   d. Byte--a sequence of binary digits acted upon as a unit--usually shorter than a computer word.

   e. Word--a set of bit that occupies one storage location--usually consisting of two bytes.

   f. Modulas--the number of discrete conditions a device or system can indicate.

   g. Complement--a quantity that completes the modulas or highest count of a system, frequently used to represent the negative of a quantity.

BINARY	OCTAL	DECIMAL	HEXIDECIMAL
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11
10010	22	18	12
10011	23	19	13
10100	24	20	14
10101	25	21	15
10110	26	22	16
10111	27	23	17
11000	30	24	18
11001	31	25	19
11010	32	26	1A
11011	33	27	1B
11100	34	28	1C
11101	35	29	1D
11110	36	30	1E
11111	37	31	1F
100000	40	32	20
100001	41	33	21
100010	42	34	22

COMPARISION OF NUMBER SYSTEMS

FIGURE 1 - Comparison of number systems.

h. Open--end math--the process by which a carry or borrow generated by the most-significant digit is disregarded.

i. Closed-end-math--the process by which a carry or borrow generated by the most-significant digit is brought to the least-significant digit and added.

j. Positional notation--a number system where the value of a digit depends upon its position.

2. Commonly used numbering systems include:

a. Decimal, radix 10, 10 symbols, 0 through 9

b. Binary, radix 2, 2 symbols, 0 and 1

c. Octal, radix 8, 8 symbols, 0 through 7

c. Hexadecimal, radix 16, 16 symbols; 0 through 9 (first 10 symbols) and A through F (last 6 symbols)

3. Positional notation

a. The value of a digit depends upon the position of the digit within a number. The value of each position is equal to the radix raised to a power.

b. The number "726_{10}" is equal to $7 \times 10^2 + 2 \times 10^1 + 6 \times 10^0$.

10^2	10^1	10^0	Power
100	10	1	Positional value in decimal
7	2	6	Number (decimal)

FIGURE 2 - Decimal structure.

c. The binary number "101_2" is equal to $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$, or $4 + 0 + 1 = 5_{10}$

2^2	2^1	2^0	Power
4	2	1	Positional value in decimal
1	0	1	Number (binary)

FIGURE 3 - Binary structure.

d. The octal number "526_8" is equal to $5 \times 8^2 + 2 \times 8^1 + 6 \times 8^0$, or $320 + 16 + 6 = 343_{10}$

8^2	8^1	8^0	Power
64	8	1	Positional value in decimal
5	2	6	Number

FIGURE 4 - Octal structure.

e. The hexadecimal number "$1A9_{16}$" is equal to the following:

$1 \times 16^2 + A(10) \times 16^1 + 9 \times 16^0$, or $256 + 160 + 9 = 425_{10}$

16^2	16^1	16^0	Power
256	16	1	Positional value in decimal
1	A(10)	9	Number (hexadecimal)

Hexadecimal structure.

4. Conversion of numbering systems is easily accomplished.

   a. Changing decimal to binary, octal, and hexadecimal is done through division. Progressively, divide the decimal by the radix of the system to which it is being converted, record the remainder after each division, and continue division until the quotient becomes 0.

   b. Convert 26_{10} to binary (radix 2).

      (1) Place in standard division form.

      $2\overline{)26}$

      (2) Divide 26 by the radix and record the remainder.

      $2\overline{)\begin{array}{r}13\\26\end{array}}$ remainder of 10

      (3) Divide the quotient, "13," by the radix and record the remainder.

      $2\overline{)\begin{array}{r}6\\13\end{array}}$ remainder of 1

      (4) Divide the quotient, "6," by the radix and record the remainder.

      $2\overline{)\begin{array}{r}3\\6\end{array}}$ remainder of 0

      (5) Divide the quotient, "3," by the radix and record the remainder.

      $2\overline{)\begin{array}{r}1\\3\end{array}}$ remainder of 1

      (6) Divide the quotient, "1," by the radix and record the remainder.

      $2\overline{)\begin{array}{r}0\\1\end{array}}$ remainder of 1

      (7) Conversion has been completed.

(8) Write the result.

$26_{10} = 11010_2$

c. Convert 105_{10} to octal (radix 8).

(1) Place in standard division form.

$8\overline{)105}$

(2) Divide 105 by the radix and record remainder.

$8\overline{)\begin{array}{r}13\\105\end{array}}$ remainder of 1

(3) Divide the quotient, "13," by the radix and record the remainder.

$8\overline{)\begin{array}{r}1\\13\end{array}}$ remainder of 5

(4) Divide the quotient, "1," by the radix and record the remainder.

$8\overline{)\begin{array}{r}0\\1\end{array}}$ remainder of 1

(5) Conversion has been completed.

(6) Write the result.

$105_{10} = 151_8$

d. Convert 573_{10} to hexadecimal (radix 16)

(1) Place in standard division form.

$16\overline{)573}$

(2) Divide 573 by the radix and record the remainder.

$16\overline{)\begin{array}{r}35\\573\end{array}}$ remainder of 13

(3) Divide the quotient, "35," by the radix and record the remainder.

$16/\overline{35}^{\,2}$ remainder of 3

(4) Divide the quotient, "2," by the radix and record the remainder.

$16/\overline{2}^{\,0}$ remainder of 2

(5) Change remainders that are larger than 9 to hexadecimal value.

$13 = D_{16}$

(6) Conversion has been completed.

(7) Write the result.

$573_{10} = 23D_{16}$

5. Conversion of binary, octal, and hexadecimal to decimal is also easily done. The process is performed through multiplication and addition. Multiply the most significant digit (MSD) of the number by the radix. Then add the product to the next digit to the right and multiply again by the radix. Continue the process until the least-significant (LSD) has been added.

a. Convert 110_2 to decimal.

(1) Write the binary number.

$1\ 1\ 0_2$

(2) Multiply MSD by the radix (2).

$$\begin{array}{r} 1 \\ \times 2 \\ \hline 2 \end{array}$$

(3) Add the product, "2," to the next digit to the right.

$$\begin{array}{r} 1 \\ +2 \\ \hline 3 \end{array}$$

(4) Multiply the sum, "3," by the radix (2).

$$\begin{array}{r} 3 \\ \times 2 \\ \hline 6 \end{array}$$

(5) Add the product, "6," to the next digit to the right.

$$\begin{array}{r} 0 \\ +6 \\ \hline \end{array}$$

(6) Conversion has been completed.

(7) Write the result.

$110_2 = 6_{10}$

b. Convert 151_8 to decimal.

(1) Write the octal number.

$1\ 5\ 1_8$

(2) Multiply the MSD by the radix (8).

$$\begin{array}{r} 1\ \ 5\ \ 1 \\ \times 8 \\ \hline 8 \end{array}$$

(3) Add the product, "8," to the next digit to the right.

$$\begin{array}{r} 5 \\ \times\ 8 \\ \hline 13 \end{array}$$

(4) Multiply the sum, "13," by the radix (8).

$$\begin{array}{r} 13 \\ \times\ \ 8 \\ \hline 104 \end{array}$$

(5) Add the product, "104," to the next digit to the right.

$$\begin{array}{r} 1 \\ +\ 104 \\ \hline 105 \end{array}$$

(6) Conversion has been completed.

(7) Write the result.

$151_8 = 105_{10}$

c. Convert $23D_{16}$ to decimal.

(1) Write the hexadecimal number.

$2\ 3\ D_{16}$

(2) Multiply the MSD by the radix (16).

$$\begin{array}{r} 2 \\ \times\ 16 \\ \hline 32 \end{array}$$

(3) Add the product, "32," to the next digit to the right.

$$\begin{array}{r} 3 \\ +32 \\ \hline 35 \end{array}$$

(4) Multiply the sum, "35," by the radix (16).

$$\begin{array}{r} 35 \\ +\ 16 \\ \hline 560 \end{array}$$

(5) Add the product, "560," to the next digit to the right.

$$\begin{array}{r} 13 \quad (D) \\ +\ 560 \\ \hline 573 \end{array}$$

Refer to figure 1 for the value of (D).

(6) Conversion has been completed.

(7) Write the result.

$23D_{16} = 573_{10}$

6. Conversion of binary related systems

   a. Converting octal to binary is performed by grouping. Use groups of three binary digits and assign each digital a binary value. Then, combine the groups.

   b. Convert 316_8 to binary (radix 2).

      (1) Write the octal number.

          $3\ 1\ 6_8$

      (2) Start with the least-significant digit.

      (3) Write the binary equivalent of each octal digit.

          3 1 6

          011 001 110

      (4) Combine the groups.

          011001110

      (5) Conversion has been completed.

      (6) Write the result.

          $316_8 = 011001110_2$

   c. Conversion of binary to octal is performed by grouping. Separate the binary into groups of three digits and then assign each group an octal equivalent. Once this has been done, combine the octal digits.

   d. Convert 101100010_2 to octal (radix 8).

      (1) Write the binary number.

          101100010_2

      (2) Start with the least-significant digit.

      (3) Separate the binary number into groups of three digits.

          101 100 010

(4) Write the octal equivalent of each group of binary digits.

e

101 100 010

5 4 2

(5) Combine the octal digits

542

(6) Conversion has been completed.

(7) Write the result.

$101100010_2 = 542_8$

e. The conversion of hexadecimal to binary is performed by using four groups of binary digits. Assign each hexadecimal digit a binary value and then combine the groups.

f. Convert 5 A_{16} to binary (radix 2).

(1) Write the hexadecimal number.

5 A_{16}

(2) Start with the least-significant digit.

(3) Write the binary equivalent of each hexadecimal digit.

5 A(16) Refer to figure 1 for the hexadecimal value of (A).

0101 1010

(4) Combine the groups.

01011010

(5) Conversion has been completed.

(6) Write the result.

$5A_{16} = 01011010_2$

g. Converting binary to hexadecimal involves separating the binary numbers into groups of four digits. Assign each group a hexadecimal equivalent and then combine the groups.

h. Convert 11111001_2 to hexadecimal (radix 16).

(1) Write the binary number.

11111001_2

(2) Start with the least-significant digit.

(3) Separate the binary number into groups of four digits.

1111 1001

(4) Write the hexadecimal equivalent of each group of binary digits.

1111 1001 Refer to figure 1 for hexadecimal equivlent.

F 9

(5) Combine the groups.

F9

(6) Conversion has been completed.

(7) Write the result.

$11111001_2 = F9_{16}$

7. Binary arithmetic

a. Addition is the direct method performed in the same manner as in decimal.

b. Fundamental rules

(1) 0 + 0 = 0 with no carry

(2) 0 + 1 = 1 with no carry

(3) 1 + 0 = 1 with no carry

(4) 1 + 1 = 0 with a carry of 1

c. Add $101_2 + 101_2$

(1) Place in column form.

1 0 1 augend
↓ addend
sum

(2) Add the first column.

```
  1
+ 1
---
  0 with a carry of 1
```

(3) Insert carry in next column and add.

```
   1
   0
 + 0
---
   1 with no carry
```

(4) Add the last column.

```
  1
+ 1
---
  0 with a carry of 1
```

(5) The last carry is the most-significant digit of the sum.

(6) Addition has been completed.

(7) Write the result of addition.

1010_2

d. Subtraction is the direct method performed in the same manner as in decimal.

e. Fundamental rules

(1) 0 - 0 = 0 with no borrow

(2) 1 - 1 = 0 with no borrow

(3) 1 - 0 = 1 with no borrow

(4) 0 - 1 = 1 with a borrow

f. Subtract $110_2 - 101_2$.

(1) Place in column form.

```
    1  1  0   minuend
- 1  0  1   subtrahend
-----------
               difference
```

(2) First column requires a borrow.

NOTE: In binary, a borrow involves borrowing 2 from the next higher order column which also decreases that column by 1.

(3) Borrow 2 from the next higher order column and place over first column.

$$\begin{array}{r} \ 0\ \ 2 \\ -\ 1\ \ 1\ \ 0 \\ \hline \end{array}$$

(4) Subtract first column.

$$\begin{array}{r} 2 \\ -1 \\ \hline 1 \end{array}$$

(5) Subtract next column.

$$\begin{array}{r} 0 \\ -0 \\ \hline 0 \end{array}$$

(6) Subtract last column.

$$\begin{array}{r} 1 \\ -1 \\ \hline 0 \end{array}$$

(7) Subtraction has been completed.

(8) Write the result of subtraction.

$0\ 0\ 1_2$

g. Multiplication is the direct method performed in the same manner as is in decimal.

h. Fundamental rules:

(1) $1 \times 1 = 1$

(2) All other combinations are equal to 0.

i. Multiply $101_2 \times 11_2$.

(1) Place in column form. Develop on the chalkboard.

```
  1  0  1   multiplicand
x    1  1   multiplier
---------
           partial products
           final product
```

(2) Multiply the multiplicand by the least-significant digit of the multiplier.

```
  1  0  1
x    1  1
---------
  1  0  1   partial products
```

(3) Multiply the multiplicand by the next digit of the multiplier.

```
    1  0  1
 x     1  1
 ----------
    1  0  1
1   0  1      partial products
```

(4) Add the partial products.

```
    1 0 1
1 0 1
-------
1 1 1 1      final product
```

(5) Multiplication has been completed.

(6) Write the result of multiplication.

1111_2

j. Division is performed by the direct method in the same manner as in decimal.

k. Fundamentals rules

(1) 1 ÷ 1 = 1

(2) 0 ÷ 1 = 0

(3) The divisor can only go into the dividend one time.

1. Divide $10010_2 \div 110$

(1) Place in standard division form.

```
      _____  quotient
110/ 10010  dividend
divisor
```

(2) Divide dividend by divisor.

```
      1
110/ 10010
```

(3) Subtract the divisor from the dividend.

```
      1
110/ 10010
     110
     ---
     110
```

(4) Divide the difference by the divisor.

```
      11
110/ 10010
     110
     ---
     110
     110
     ---
       0
```

(5) Division has been completed.

(6) Write the result of division.

11_2

8. Complement arithmetic

a. The laws of decimal addition and subtraction apply to complement arithmetic. Subtraction and addition are performed by addition. This is possible since A + B = SUM and A + (-B) = DIFFERENCE. Values are represented in binary with the most-significant bit of the complement being the sign bit.

(1) A binary 1 indicates a negative value.

(2) A binary 0 indicates a positive value.

(3) All math functions can be performed by adders, which requires fewer circuits.

b. Types of complements

(1) The 1's complement is used in the addition of signed numbers; subtraction is performed by addition. The rules for complementation are as follows:

(a) Change each binary 1 to a 0.

(b) Change each binary 0 to 1.

```
00010101 BINARY NUMBER

11101010 1's COMPLEMENT FORM
```

FIGURE 6 - The 1's complement.

(c) The 1's complement of a number, using eight bits:

00001100_2 represents a +12

11110011_2 represents a -12

NOTE: The most-significnt bit is a 1, indicating a negative number.

(2) The 2's complement is used in the addition of signed numbers; subtraction is performed by addition. The rules for complementation are as follows:

(a) Develop the 1's complement of the binary number.

(b) Add 1 to the 1's complement.

```
00010101 BINARY NUMBER
11101010 1's COMPLEMENT
      +1 ADD 1
--------
11101011 2's COMPLEMENT FORM
```

FIGURE 7 - The 2's complement.

(c) The 2's complement representation of a number, using eight bits

00001100_2 represents a +12

11110100_2 represents a -12

NOTE: The most-significant bit is a 1 indicating a negative number.

c. The 1's complement method of addition is used with closed-end math and the rules are as follows:

(1) Negative numbers are shown in the 1's complement form.

(2) Add the numbers, including the sign bit.

(3) When a carry is generated by the sign-bit column, perform an end-around and add.

(4) A negative answer will be in the 1's complement form.

(5) A positive answer will be in true binary form.

d. Add $(+13_{10}) + (11_{10})$

NOTE: Use 8-bit numbers for simplicity.

(1) Both numbers are positive; no complementation is required.

(2) Write the binary value of each number.

+13 = 00001101
+11 = 00001011

(3) Add the addend to the augend.

```
 00001101
+00001011
 00011000
```

(4) No carry was generated by the sign-bit column.

(5) The result is a $+24_{10}$ in true binary form.

00011000

e. Add $(-13_{10}) + (-11_{10})$

(1) Write the 1's complement of each number.

-13 = 11110010
-11 = 11110100

(2) Add the complemented addend to the complemented augend.

```
  1  carry
    110010
+11110100
 11100110
```

(3) A carry was generated by the sign bit column.

(4) Perform the end-around carry and add.

```
 11110010
+11110100
 11100110
+        1
 11100111
```

(5) The result is a -24_{10} in the 1's complement form.

11100111

f. Add $(-13_{10}) + (11_{10})$

(1) Write -13 in the 1's complement form.

-13 = 11110010

(2) Write +11 in true binary form.

+11 = 00001011

(3) Add the addend to the complementd augend.

```
 11110010
+00001011
 11111101
```

(4) No carry was generated by the sign-bit column.

```
110010
001011
111101
```

(5) The result is a -2_{10} in the 1's complement form.

11111101

g. Add $(-11_{10}) + (13_{10})$

(1) Write -11 in the 1's complement form.

-11 = 11110100

(2) Write +13 in true binary form.

+13 = 00001101

(3) Add the addend to the complemented augend.

```
  1  carry
 11110100
+00001101
 00000001
```

(4) A carry was generated by the sign-bit column.

(5) Perform the end-around carry and add.

```
 11110100
+00001101
 00000001
        1
 --------
 00000010
```

(6) The result is $+2_{10}$ in true binary form.

00000010

h. The 1's complement method of subtraction is used with closed-end math. The rules are as follows:

(1) Complement the subtrahend, including the sign bit, and add.

(2) Apply the rules for 1's complement.

(3) Negative numbers are shown in the 1's complement form.

i. Subtract $(+13_{10}) - (+11_{10})$.

(1) Write the binary value of each number.

+13 = 00001101
+11 = 00001011

(2) Form the 1's complement of the subtrahend.

```
00001011
11110100
```

(3) Add the complemented subtrahend to the minuend.

```
 1  carry
 00001101
+11110100
 00000001
```

(4) A carry was generated by the sign-bit column.

(5) Perform the end-around carry and add.

```
 00001101
+11110100
 00000001
+        1
 00000010
```

(6) The result is a $+2_{10}$ in true binary form.

```
00000010
```

j. Subtract $(-13_{10}) - (-11_{10})$

(1) Write the 1's complement of each number.

```
-13 = 11110010
-11 = 11110100
```

(2) Form the 1's complement of the subtrahend.

```
11110100
00001011
```

(3) Add the complemented subtrahend to the minuend.

```
 11110010
+00001011
 11111101
```

(4) No carry was generated by the sign-bit column.

(5) The result is a -2_{10} in 1's complement form.

```
11111101
```

k. Subtract $(-13_{10}) - (+11_{10})$

(1) Write -13 in the 1's complement form.

-13 = 11110010

(2) Write +11 in true binary form.

+11 = 00001011

(3) Form the 1's complement of the subtrahend.

```
00001011
11110100
```

(4) Add the complemented subtrahend to the minuend.

```
 1  carry
 11110010
+11110100
 11100110
```

(5) A carry was generated by the sign-bit column.

(6) Perform the end-around carry and add

```
 11110010
+11110100
 11100110
+       1
 11100111
```

(7) The result is a -24_{10} in 1's complement form.

11100111

l. Subtract $(+13_{10}) - (-11_{10})$

(1) Write +13 in true binary form.

+13 = 00001101

(2) Write -11 in 1's complement form.

-11 = 11110100

(3) Form the 1's complement of the subtrahend.

```
11110100
00001011
```

(4) Add the complemented subtrahend to the minuend.

```
 00001101
+00001011
 00011000
```

(5) No carry was generated by the sign-bit column.

(6) The result is a $+24_{10}$ in true binary form.

00011000

1. The 2's complement method of addition is used with open-end math.

NOTE: The COM-TRAN Ten uses the 2's complement method of addition and subtraction. The rules are as follows:

(1) Add the numbers, including the sign bit.

(2) A carry generated by the sign-bit column is disregarded.

(3) A negative answer will be in the 2's complement form.

(4) A positive answer will be in true binary form.

(5) Negative numbers are shown in the 2's complement form.

1. Add $(+13_{10}) + (+11_{10})$.

NOTE: Use 8-bit binary numbers for simplicity.

(1) Both numbers are positive; no complementation is required.

(2) Write the binary value for each number.

```
+13 = 00001101
+11 = 00001011
```

(3) Add the addend to the augend.

```
 00001101
 00001011
 00011000
```

(4) No carry was generated by the sign-bit column.

(5) The result is a $+24_{10}$ in true binary form.

00011000

o. Add $(-13_{10}) + (-11_{10})$

(1) Write the 2's complement of each number.

-13 = 11110011
-11 = 11110101

(2) Add the complemented addend to the complemented augend.

```
 1  carry
 11110011
+11110101
 11101000
```

(3) Disregard the carry from the sign-bit column.

(4) The result is a -24_{10} in the 2's complement form.

11101000

p. Add $(+13_{10}) + (-11_{10})$

(1) Write +13 in true binary form.

+13 = 00001101

(2) Write -11 in the 2's component form.

-11 = 11110101

(3) Add the complemented addend to the augend.

```
 1  carry
 00001101
 11110101
 00000010
```

(4) Disregard the carry from the sign-bit column.

(5) The result is a $+2_{10}$ in true binary form.

00000010

q. Add $(-13_{10}) + (+11_{10})$.

(1) Write the -13 in the 2's complement form.

-13 = 11110011

(2) Write the +11 in true binary form.

+11 = 00001011

(3) Add the addend to the complemented augend.

```
 11110011
+00001011
 11111110
```

(4) No end carry was generated by the sign-bit column.

(5) The result is a -2_{10} in the 2's complement form.

11111110

r. The 2's complement method of subtraction is used with open-ended math. The rules are as follows:

(1) Complement the subtrahend, including the sign bit and add.

(2) Apply the rules for 2's complement addition.

(3) Negative numbers are shown in the 2's complement form.

s. Subtract $(+13_{10}) - (+11_{10})$.

(1) Write the binary value for each number.

+13 = 00001101
+11 = 00001011

(2) Determine the 2' complement of the subtrahend.

00001011
11110101

(3) Add the complemented subtrahend to the minuend.

```
 1  carry
 00001101
+11110101
```

(4) Disregard the carry from the sign-bit column.

(5) The result is a $+2_{10}$ in true binary form.

00000010

t. Subtract $(-13_{10}) - (-11_{10})$.

(1) Write the 2's complement of each number.

-13 = 11110011
-11 = 11110101

(2) Form the 2's complement of the subtrahend.

11110101
00001011

(3) Add the complemented subtrahend to the minuend.

11110011
00001011
11111110

(4) No carry was generated by the sign-bit column.

(5) The result is a -2_{10} in the 2's complement form.

11111110

u. Subtract $(-13_{10}) - (+11_{10})$.

(1) Write -13 in the 2's complement form.

-13 = 11110011

(2) Write +11 in true binary form.

+11 = 00001011

(3) Form the 2's complement of the subtrahend.

00001011
11110101

(4) Add the complemented subtrahend to the minuend.

1 carry
11110011
11110101
11101000

(5) Disregard the carry from the sign-bit column.

(6) The result is a -24_{10} in the 2's complement form.

11101000

v. Subtract $(+13_{10}) - (-11_{10})$.

(1) Write +13 in true binary form.

+13 = 00001101

(2) Write -11 in the 2's complement form.

-11 11110101

(3) Form the 2's complement of the subtrahend.

11110101
00001011

(4) Add the complemented subtrahend to the minuend.

```
 00001101
+00001011
 00011000
```

(5) No carry was generated by the sign-bit column.

(6) The result is a $+24_{10}$ in true binary form.

00011000

9. Computer multiplication is calculated by three different methods.

a. Multiplication by repetitive addition

(1) Requires a counter to keep track of additions.

(2) Disadvantages is that it requires too much time for large numbers.

b. Multiplication by shifting left

(1) In binary, this is the equivalent of multiplying by 2.

(2) Disadvantage is that multiplication is limited to powers of 2.

c. Multiplication by addition and shift right

(1) Most common method used in digital computers.

(2) Advantage is that it requires less time to perform.

(3) The COM-TRAN Ten uses the addition and shift right method.

10. Multiplication, addition, and shift right

a. The names and purposes of the registers used in the operation are as follows (refer to figure 8; note that this is a simplified algorithm flow chart of the COM-TRAN Ten).

(1) Accumultor register (A-register)

(a) Holds the result of each addition.

(b) Holds the two high-order bytes of the product after multiplication has been completed.

(2) Buffer register (B-register) holds the multiplicand during multiplication.

(3) Quotient register (Q register)

(a) Holds the multiplier prior to multiplication.

(b) Holds the two low-order bytes of the product after multiplication has been completed.

(4) Countdown register (C-register)

(a) Holds the number of shifts.

(b) Number of shifts is the computer-word length.

(c) COM-TRAN Ten word length is 8 bits.

b. Multiplication operation of the COM-TRAN Ten

NOTE: First four steps of the flow chart set up the registers.

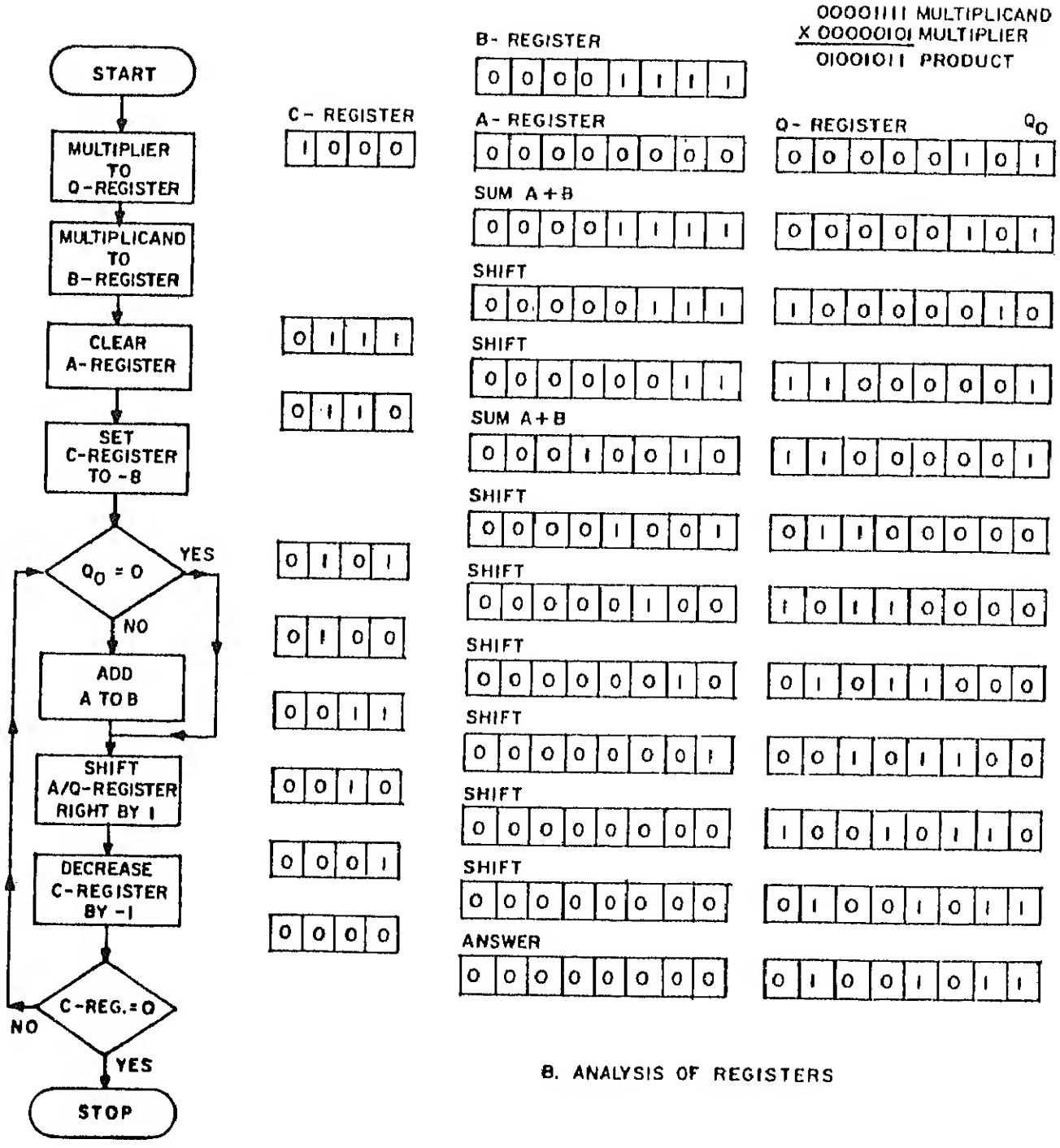
(1) Transfers multiplier from memory to the Q-register: Q-register contains 00000101.

(2) Transfers multiplicand from A-register to B-register: B-register contains 00001111.

(3) Clears the A-register: A-register contains 00000000.

(4) Sets C-register to a count of 8: C-register contains 1000.

NOTE: C-register in the COM-TRAN Ten is 8 bits.



A. ALGORITHM FLOW CHART

B. ANALYSIS OF REGISTERS

FIGURE 8

MULTIPLICATION BY ADDITION AND SHIFT RIGHT

NOTE: Addition will no longer take place. The Q0 bit will remain 0; only the shifting of the AQ-register right will take place.

(20) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000100.

(b) Q-register now contains 10110000.

(21) Decreases C-register by 1: C-register now contains 0100.

(22) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(23) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(24) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000010.

(b) Q-register now contains 01011000.

(25) Decreases C-register by 1: C-register now contains 0011.

(26) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(27) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

Shifts contents of AQ-register right one place.

(a) A-register now contains 00000001.

(b) Q-register now contains 00101100.

Decreases C-register by 1: C-register now contains 0010.

A decisión is made.

(a) C-register equals 0.

(b) Decision is no.

A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

Shifts contents of AQ-register right one place.

(a) A-register now contains 00000000.

(b) Q-register now contains 10010110.

Decreases C-register by 1: C-register now contains 0001.

A decision is made.

(a) C-register equals 0.

(b) Decision is no.

A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

Shifts contents of AQ-register right one place.

(a) A-register now contains 00000000.

(b) Q-register now contains 01001011.

NOTE: Addition will no longer take place. The c
bit will remain 0; only the shifting of th
AQ-register right will take place.

(20) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000100.

(b) Q-register now contains 10110000.

(21) Decreases C-register by 1: C-register now contains 0100.

(22) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(23) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(24) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000010.

(b) Q-register now contains 01011000.

(25) Decreases C-register by 1: C-register now contains 0011.

(26) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(27) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(28) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000001.

(b) Q-register now contains 00101100.

(29) Decreases C-register by 1: C-register now contains 0010.

(30) A decisión is made.

(a) C-register equals 0.

(b) Decision is no.

(31) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(32) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000000.

(b) Q-register now contains 10010110.

(33) Decreases C-register by 1: C-register now contains 0001.

(34) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(35) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(36) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000000.

(b) Q-register now contains 01001011.

NOTE: Addition will no longer take place. The Q0 bit will remain 0; only the shifting of the AQ-register right will take place.

(20) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000100.

(b) Q-register now contains 10110000.

(21) Decreases C-register by 1: C-register now contains 0100.

(22) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(23) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(24) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000010.

(b) Q-register now contains 01011000.

(25) Decreases C-register by 1: C-register now contains 0011.

(26) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(27) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(28) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000001.

(b) Q-register now contains 00101100.

(29) Decreases C-register by 1: C-register now contains 0010.

(30) A decisión is made.

(a) C-register equals 0.

(b) Decision is no.

(31) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(32) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000000.

(b) Q-register now contains 10010110.

(33) Decreases C-register by 1: C-register now contains 0001.

(34) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(35) A decision is made.

(a) Q0 bit equals 0.

(b) Decision is yes.

(c) No addition takes place.

(36) Shifts contents of AQ-register right one place.

(a) A-register now contains 00000000.

(b) Q-register now contains 01001011.

(37) Decreases C-register by 1: C-register now contains 0000.

(38) A decision is made.

(a) C-register equals 0.

(b) Decision is yes.

(39) Stops, multiplication has been completed.

(40) The product is displayed in the AQ-register:

00000000 01001011

11. Computer division is done by three different methods.

a. Division by repetitive subtraction

(1) Requires a counter to keep track of subtractions.

(2) Disadvantages is that too much time is required for larger numbers.

b. Division by shifting right

(1) In binary, this is the equivalent of dividing by 2.

(2) Disadvantage is that division is limited to powers of 2.

c. Division by shift left and subtract

(1) Most common method used in digital computers.

(2) Advantage is that it requires less time to perform.

(3) The COM-TRAN Ten uses the shift-left-and-subtract method.

12. Division, shift left and subtract

a. The names and purposes of the registers used in the operation are as follows (refer to figure 9; note that this is a simplified algorithm flow chart of the COM-TRAN Ten).

(1) Accumulator register (A-register)

(a) Holds the result of subtraction when the result is positive.

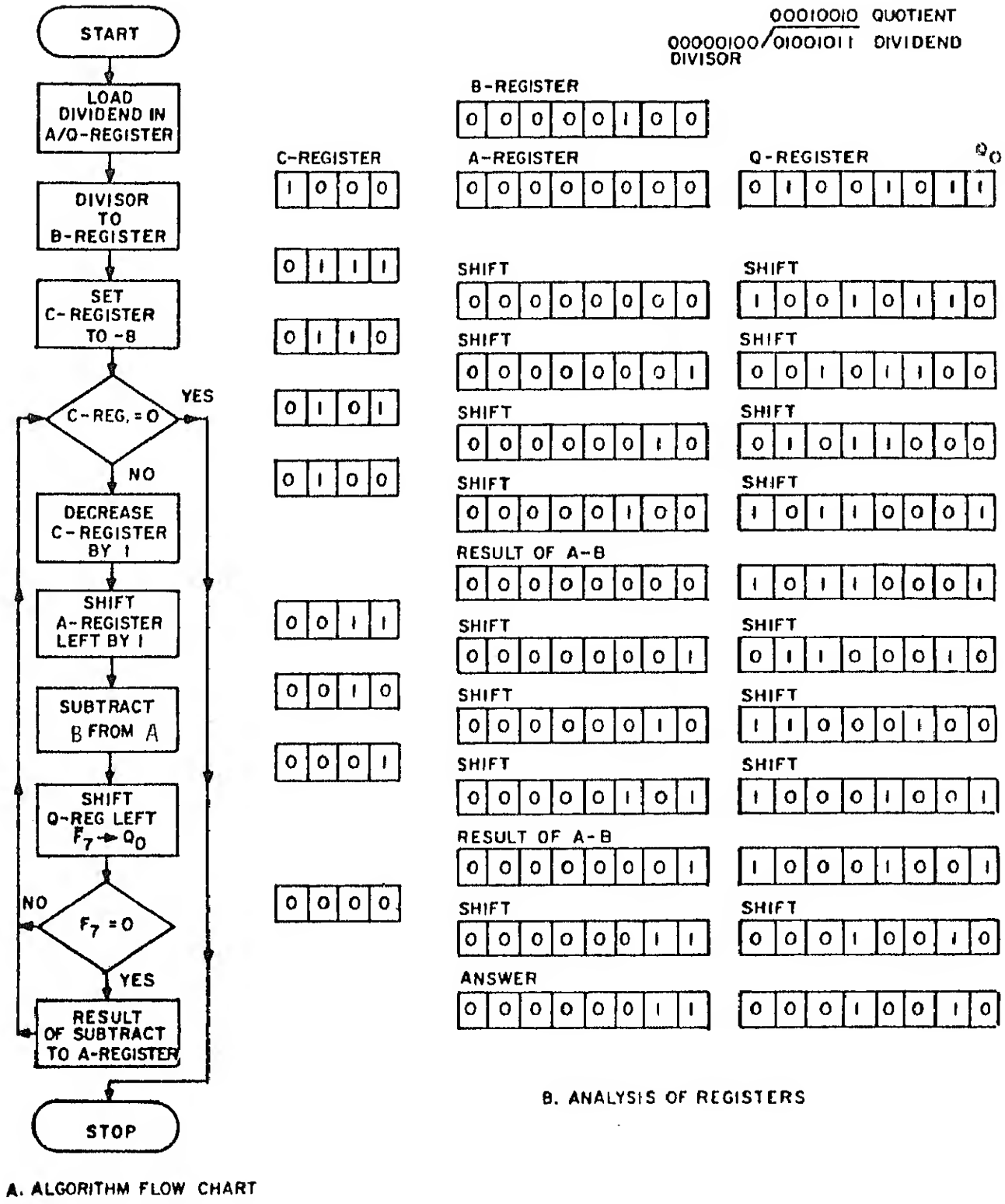


FIGURE 9 - Division by shift left and subtract.

(b) Holds the two high-order bytes of dividend prior to division.

(c) Holds the remainder immediately after division

(2) Buffer register (B-register) holds the divisor during division.

(3) Quotient register (Q-register)

(a) Holds the two low-order bytes of the dividend prior to division.

(b) Holds the quotient immediately after division.

(4) Countdown register (C-register)

(a) Holds the number of shifts.

(b) Number of shifts is the computer-word length.

(c) COM-TRAN Ten word length is 8 bits.

(5) Accumulator and quotient register (AQ-register)

(a) The A- and Q-registers are combined as one register during divide.

(b) The dividend is held in the AQ-register prior to division.

b. Division operation

NOTE: The first three steps of the flow chart set up the registers.

(1) Dividend is loaded in the AQ-register.

(a) AQ-register contains 00000000 01001011.

(b) A-register contains 00000000, the two high-order bytes of the dividend.

(c) Q-register contains 01001011, the two low-order bytes of the dividend.

(2) Transfer the divisor from memory to the B-register. B-register contains 00000100.

(3) Set the C-register to a count of 8: C-register contains 1000.

NOTE: C-register in the COM-TRAN Ten is 8 bits.

(4) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(c) When the C-register equals 0, division has been completed and the computer will stop.

(5) Decreases C-register by 1: C-register now contains 0111.

(6) Shift contents of A-register left one place.

(a) The most-significant bit of the Q-register (Q7) is shifted into the least-significant bit of the A-register (A0).

(b) A-register now contains 00000000.

(7) Subtract the contents of the B-register from the A-register.

NOTE: The 2's complement method of subtraction is used in the COM-TRAN Ten.

(a) The result of subtraction is negative (11111100).

(b) The most-significant bit of the result of subtraction is called the F7 bit.

(c) The F7 bit is a 1 when the result of the subtraction is negative.

(d) $\overline{F7}$ (F7 not) is the complement of the F7 bit, F7 is a 0 when the result of the subtraction is negative.

(8) Shift contents of Q-register left one place.

(a) The most-significant bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 0.

(c) A "0" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 10010110.

(9) A decision is made.

(a) F7 bit equals 0.

(b) Decision is no.

(c) The result of subtraction is negative and is not transferred to the A-register.

(10) A decision is made.

(a) C-reister equals 0.

(b) Decision is no.

(11) Decrease C-register by 1: C-register now contins 0110.

(12) Shift contents of A-register left one place.

(a) The Q7 bit of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000001.

(13) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is negative (11111101).

(b) F7 equals 1.

(c) $\overline{F7}$ equals 0.

(14) Shift contents of Q-register left one place.

(a) The most-significant bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 0.

(c) A "0" is shifted into the least significant bit of the Q-register (Q0).

(d) Q-register now contains 00101100.

(15) A decision is made.

(a) F7 bit equals 0.

(b) Decision is no.

(c) The result of subtraction is negative and is not transferred to the A-register.

(16) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(17) Decrease C-register by one: C-register now contains 0101.

(18) Shift contents of A-register left one place.

(a) The Q7 bi of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000010.

(19) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is negative (11111110).

(b) F7 equals 1.

(c) $\overline{F7}$ equals 0.

(20) Shift contents of Q-register left one place.

(a) The most-significant bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 0.

(c) A "0" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 01011000.

(21) A decision is made.

(a) F7 bit equals 0.

(b) Decision is no.

(c) The result of subtraction is negative and is not transferred to the A-register.

(22) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(23) Decrease C-register by 1: C-register now contains 0100.

(24) Shift contents of A-register left one place.

(a) The Q7 bit of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000100.

(25) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is positive (00000000).

(b) F7 equals 0.

(c) $\overline{F7}$ equals 1.

(26) Shift contents of Q-register left one place.

(a) The most-significnt bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 1.

(c) A "1" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 10110001.

(27) A decision is made.

(a) F7 bit equals 0.

(b) Decision is yes.

(c) The result of subtraction is positive.

(28) Transfer the result of subtraction to the A-register. A-register no contains 00000000.

(29) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(30) Decrease the C-register by 1: C-register now contains 0011.

(31) Shift content of A-register left one place.

(a) The Q7 bit of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000001.

(32) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is negative (11111101).

(b) F7 equals 1.

(c) $\overline{F7}$ equals 0.

(33) Shift contents of Q-register left one place.

(a) The most-significnt bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 0.

(c) A "0" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 01100010.

(34) A decision is made.

(a) F7 bit equals 0.

(b) Decision is no.

(c) The result of subtraction is negative and is not transferred to the A-register.

(35) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(36) Decrease the C-register by 1: C-register now contains 0010.

(37) Shift content of A-register left one place.

(a) The Q7 bit of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000010.

(38) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is negative (11111110).

(b) F7 equals 1.

(c) $\overline{F7}$ equals 0.

(39) Shift contents of Q-register left one place.

(a) The most-significnt bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 0.

(c) A "0" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 11000100.

(40) A decision is made.

(a) F7 bit equals 0.

(b) Decision is no.

(c) The result of subtraction is negative and is not transferred to the A-register.

(41) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(42) Decrease the C-register by 1: C-register now contains 0001.

(43) Shift content of A-register left one place.

(a) The Q7 bit of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000101.

(44) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is positive (00000001).

(b) F7 equals 1.

(c) $\overline{F7}$ equals 0.

(45) Shift contents of Q-register left one place.

(a) The most-significant bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 1.

(c) A "1" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 10001001.

(46) A decision is made.

(a) F7 bit equals 0.

(b) Decision is yes.

(c) The result of subtraction is positive.

(47) Transfer the result of subtraction to the A-register. A-register now contains 00000001.

(48) A decision is made.

(a) C-register equals 0.

(b) Decision is no.

(49) Decrease the C-register by 1: C-register now contains 0000.

(50) Shift content of A-register left one place.

(a) The Q7 bit of the Q-register is shifted into the A0 bit of the A-register.

(b) A-register now contains 00000011.

(51) Subtract the contents of the B-register from the A-register.

(a) The result of subtraction is negative (11111111).

(b) F7 equals 1.

(c) $\overline{F7}$ equals 0.

(52) Shift contents of Q-register left one place.

(a) The most-significant bit of the Q-register (Q7) is dropped.

(b) $\overline{F7}$ equals 1.

(c) A "0" is shifted into the least-significant bit of the Q-register (Q0).

(d) Q-register now contains 00010010.

(53) A decision is made.

(a) F7 bit equals 0.

(b) Decision is no.

(c) The result of subtraction is negative and is not transferred to the A-register.

(54) A decision is made.

(a) C-register equals 0.

(b) Decision is yes.

(c) Stops, division has been completed.

BASE 10	PURE BINARY	GRAY CODE
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

FIGURE 10 - Comparison of digital, binary, and Gray code.

(55) The quotient is displayed in the Q-register: 00010010.

(56) The remainder is displyed in the A-register: 00000011.

13. Gray code

a. Advantages include the reduction of operational errors. Gray-coded numbers change only one digit with each successive change in value. This property reduces the chance of errors.

b. Gray code is commonly used in digital encoding circuits and the COM-TRAN Ten uses Gray code to initiate the clock pulses.

14. Conversion of binary to Gray code and the conversion of binary to Gray code (refer to figure 10).

a. Binary to Gray code

(1) Rules

(a) The most-significant digit of the binary number will be the most-significant digit of the Gray code number.

(b) Add the binary digits horizontally, starting with the most-significant digit.

(c) The sum of each addition is the next Gray code digit.

(d) Disregard any carries.

(e) Continue until the least-significant digit is reached.

(2) Convert 0110_2.

(a) Write the binary number.

0 1 1 0

(b) Write the (MSD) of the binary numbers as the (MSD) of the Gray code number.

0 1 1 0 binary
0 Gray code

(e) Continue until the least-significnt digit is reached.

(2) Convert 0101 to Gray code. Develop on the chalkboard.

(a) Write the Gray code number.

0 1 0 1

(b) Write the (MSD) of the Gray code number as the (MSD) of the binary number.

```
0 1 0 1   Gray code
0         binary
```

(c) Add the (MDS) of the binary number to the next Gray code digit.

0 + 1 = 1

(d) Write the sum as the next binary digit.

```
0 1 0 1   Gray code
0 1       binary
```

(e) Add the second binary digit to the third digit of the Gray code number.

1 + 0 = 1

(f) Write the sum as the next binary digit.

```
0 1 0 1   Gray code
0 1 1     binary
```

(g) Add the third binary digit to the last Gray code digit, disregard the carry.

1 + 1 = 0

(h) Write the sum as the next binary digit.

```
0 1 0 1   Gray code
0 1 1 0   binary
```

(i) Conversion has been completed.

(j) Write the result.

0110 binary

NOTETAKING SHEET 5.3.1N

BASIC LOGIC GATE INTERPRETATION

REFERENCES:

1. William, Gerald. Digital Technology. Chicago: Science Research Association, 1982. Chapter 1, pp. 5-26, Chapter 2, pp. 33-51; chapter 4, pp. 109-136; chapter 5, pp. 147-170.

2. Fluke, John. Digital Logic Fundamentals. Merrill Publishing Company, 1977. Chapter 4, pp. 67-94; chapter 8, pp. 223-256; chapter 14, pp. 460-462.

3. American National Standard Graphic Symbols for Logic Diagrams. Institute of Electrical and Electronics Engineers, 1973, ANSI Y32.14, IEEE STD 91. MIL-STD-806C. Sections 1-6.

NOTETAKING OUTLINE

I. Definitions

II. Logic Levels

A. Positive logic

POSITIVE LOGIC

Logic 1 = +5 V = HIGH = H

Logic 0 = 0 V = LOW = L

FIGURE 1 - Logic level representation.

B. Negative logic

NEGATIVE LOGIC

Logic 1 = 5 V = HIGH = H
Logic 0 = 0 V = LOW = L

FIGURE 2 - Logic level representation.

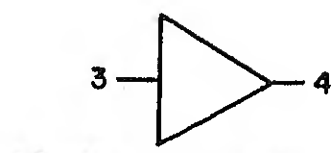
III. Negation Indicator System (logic)

A. Description

B. Application

IV. Logic Gates and Functions

A. Amplifier



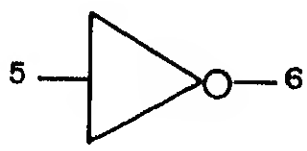
3A. AMPLIFIER SYMBOL

3	4
L	L
H	H

3B. TRUTH TABLE

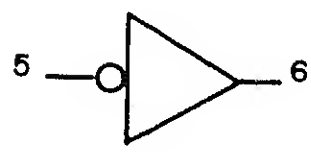
FIGURE 3 - Amplifiers logic symbol and truth table.

B. Inverter



5	6
L	H
H	L

4A Standard inverter symbol and truth table

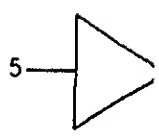


5	6
L	H
H	L

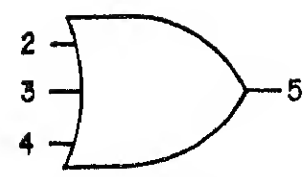
4B Alternate inverter symbol and truth table

FIGURE 4 - Inverter symbol and truth tables.

D. OR Gate



5A.

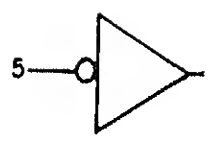


8A OR SYMBOL

2	3	4	5
L	L	L	L
L	L	H	H
L	H	L	H
H	L	L	H
H	H	H	H

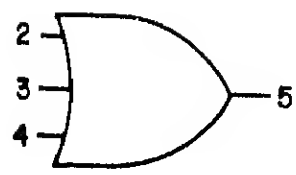
8B TRUTH TABLE

FIGURE 8 - Standard OR symbol and truth table



5B. Al

FIG



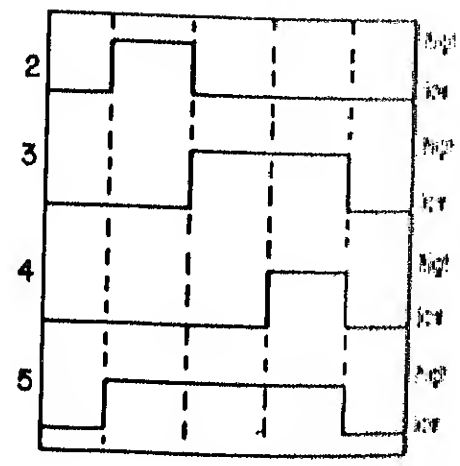
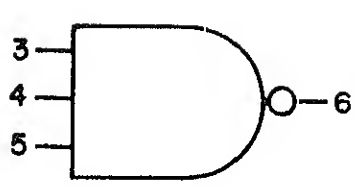


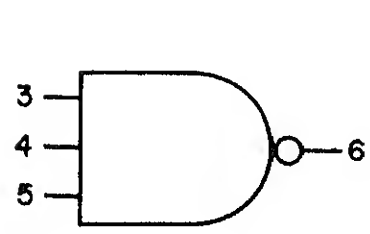
FIGURE 9 - OR symbol and timing diagram

E. NAND gate



3	4	5	6
L	L	L	H
L	L	H	H
L	H	L	H
H	L	L	H
H	H	H	L

FIGURE 10 - Standard NAND symbol and truth table.



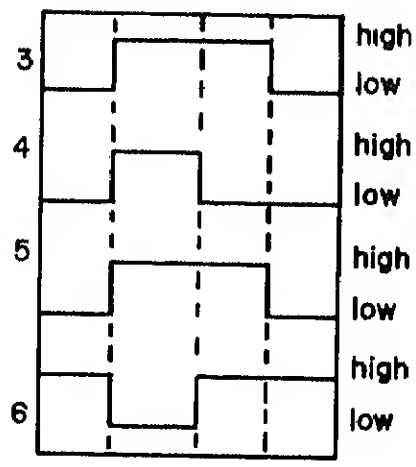
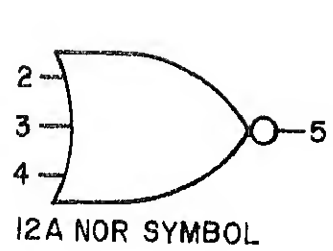


FIGURE 11 - NAND symbol and timing diagram.

F. NOR gate



12A NOR SYMBOL

2	3	4	5
L	L	L	H
L	L	H	L
L	H	L	L
H	L	L	L
H	H	H	L

12B TRUTH TABLE

FIGURE 12 - Standard NOR symbol and truth table.

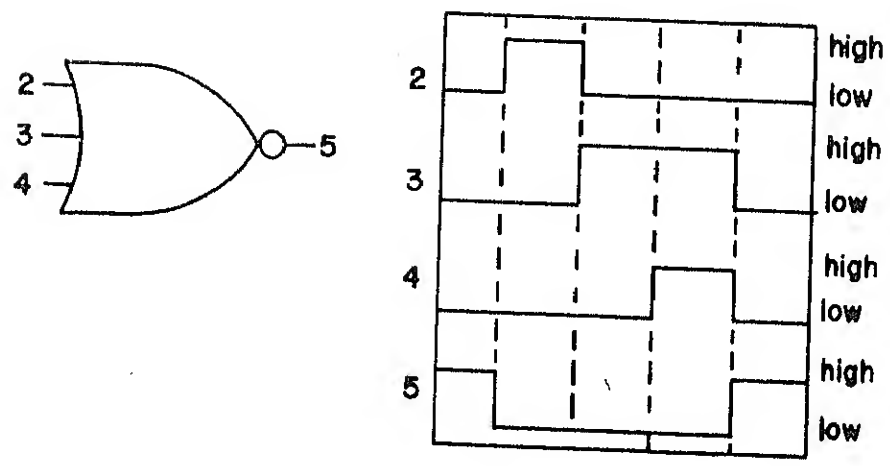
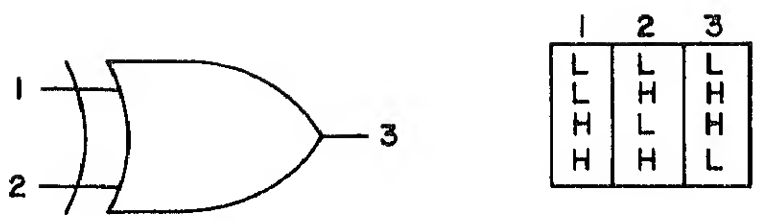


FIGURE 13 - NOR symbol and timing diagram.

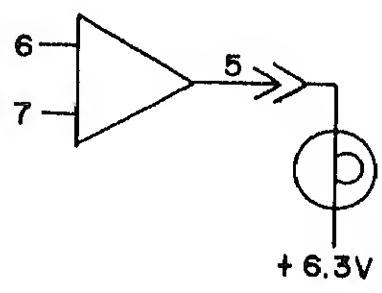
F. Exclusive OR



1	2	3
L	L	L
L	H	H
H	L	H
H	H	L

FIGURE 14 - Standard EXCLUSIVE OR symbol and truth table.

G. Positive AND driver



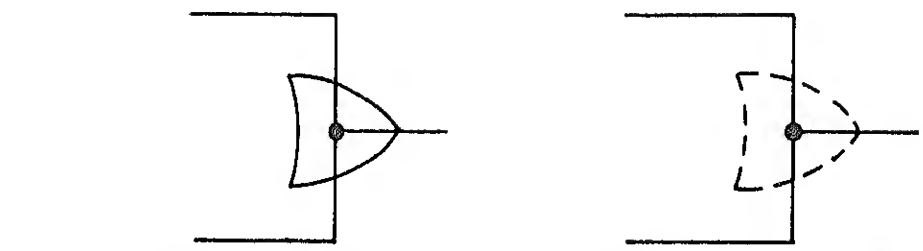
5A. POSTIVE AND DRIVER AND LAMP

6	7	5	LAMP
L	L	L	ON
L	H	L	ON
H	L	L	ON
H	H	H	OFF

15B. TRUTH TABLE

FIGURE 15 - Positive AND driver symbol and truth table.

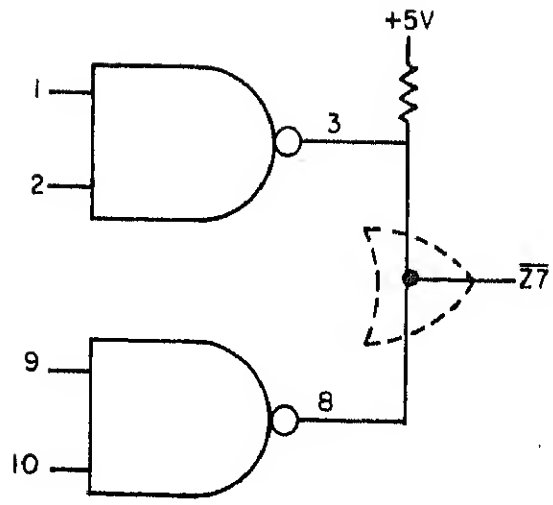
H. Wired OR



16A. MIL. STD SYMBOL

16B. PHANTON OR SYMBOL

FIGURE 16 - Wired OR symbols.



17A. WIRED OR

1	2	9	10	$\overline{Z7}$
L	L	L	L	H
L	L	H	H	L
H	H	L	L	L
H	H	H	H	L

17B. TRUTH TABLE

FIGURE 17 - Wired OR configuration and truth table.

V. Logic Gate Equivalents

A. De Morgan's theorem (law of duality)

B. NAND gate and logic dual

C. NOR gate and logic dual

D. AND gate and logic dual

VI. Flip-flops and Latches

A. General information

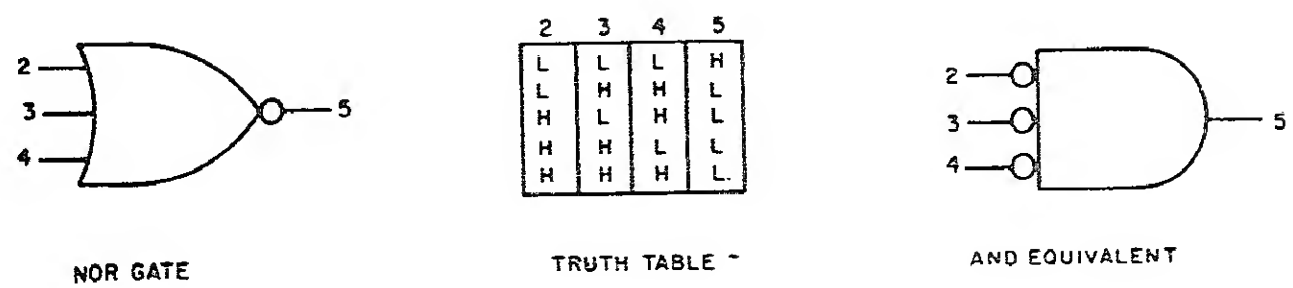


FIGURE 19 - NOR gate and AND equivalent.

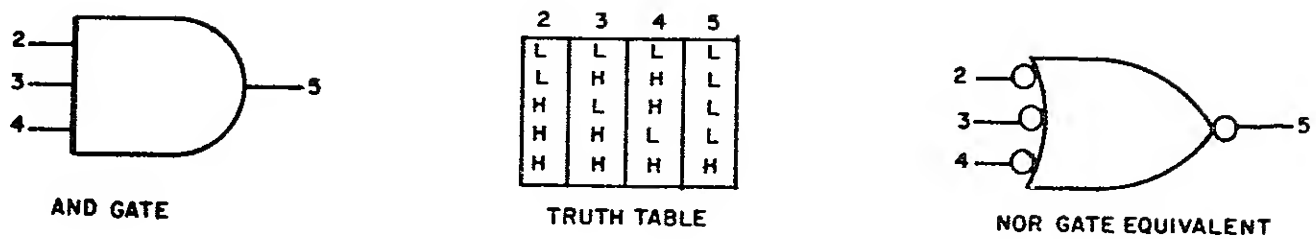
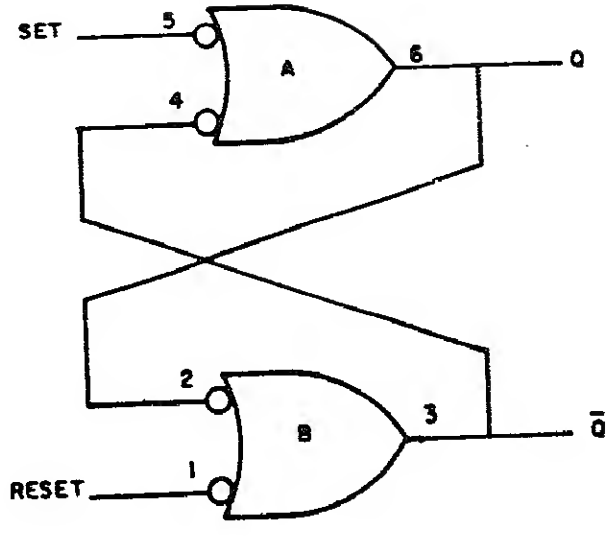
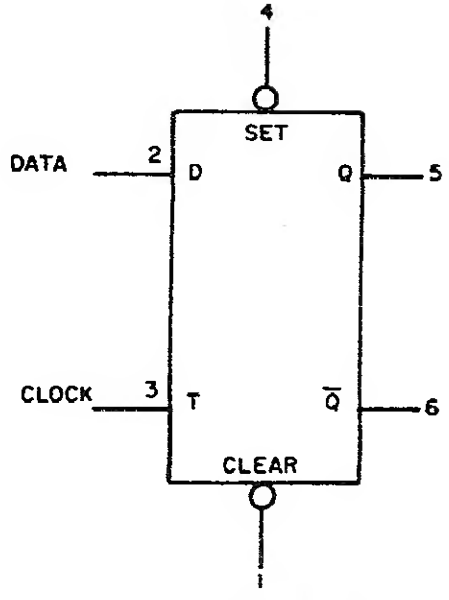


FIGURE 20 - AND gate and NOR equivalent.



INPUTS		OUTPUTS		
SET	RESET	Q	$\overline{Q}$	
LOW	HIGH	HIGH	LOW	SET STATE
HIGH	LOW	LOW	HIGH	RESET STATE

FIGURE 21 - R-S latch and truth table.



LOGIC SYMBOL

DATA	CLOCK	Q	$\overline{Q}$
LOW	⌐¬	LOW	HIGH
HIGH	⌐¬	HIGH	LOW

DATA AND CLOCK TRUTH TABLE

SET	CLEAR	Q	$\overline{Q}$
LOW	LOW	FORBIDDEN	
LOW	HIGH	HIGH	LOW
HIGH	LOW	LOW	HIGH
HIGH	HIGH	NORMAL CLOCK OPERATION USING (D) INPUT	

STATIC STATE → SET AND CLEAR

SET AND CLEAR TRUTH TABLE

FIGURE 22 - Type "D" edge-triggered flip-flop and truth table.

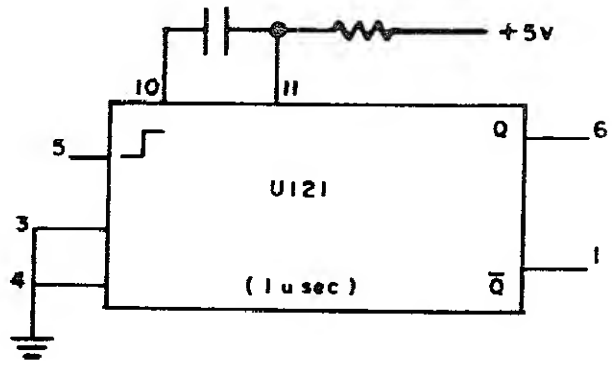
B. R-S latch

C. Type "D" edge-triggered flip-flop

VII. Monostable Multivibrator (single-shot)

A. General information

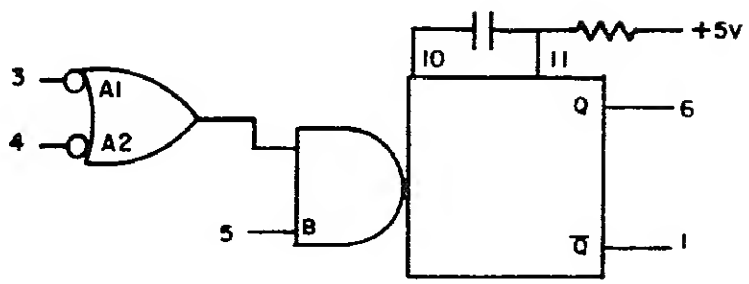
B. Single-shot operation



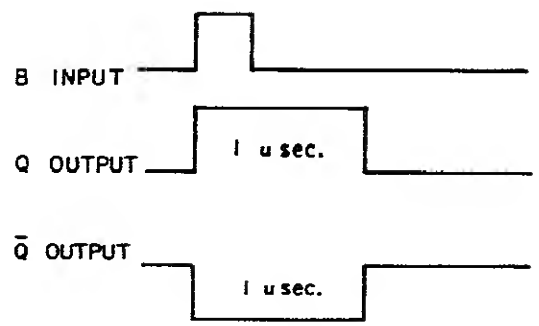
SINGLE SHOT LOGIC SYMBOL

	A1 PIN 3	A2 PIN 4	B PIN 5	Q PIN 6	Q̄ PIN 1
STATIC CONDITION	HIGH	HIGH	LOW	LOW	HIGH
A1 TRIGGER INPUT	LOW	HIGH	HIGH	HIGH	LOW
A2 TRIGGER INPUT	HIGH	LOW	HIGH	HIGH	LOW
B TRIGGER INPUT	LOW	LOW	HIGH	HIGH	LOW

TRUTH TABLE



74121 I. C. MONOSTABLE (ONE SHOT)



TIMING DIAGRAM

FIGURE 23 - Monostable multivibrator (single shot) and truth table.

NOTETAKING SHEET 5.4.1N

INTRODUCTION TO THE COM-TRAN Ten COMPUTER AND ORGANIZATION

REFERENCE:

Digiac Corporation. Technical Operations Manual for the COM-TRAN Ten M104, Vol. I. Section III, pp. 13-36; section IV, pp. 37-127.

NOTETAKING OUTLINE

I. Operating Characteristics

A. Memory Unit

B. Arithmetic unit

C. Control Unit

D. Input section

E. Output section

F. Teletypewriter

G. Logic circuit boards

II. Control Panel

A. Power and lamp test section

B. Input section

C. I/O mode section

D. Control section

E. Mode section

F. Error bypass section

G. I/O section

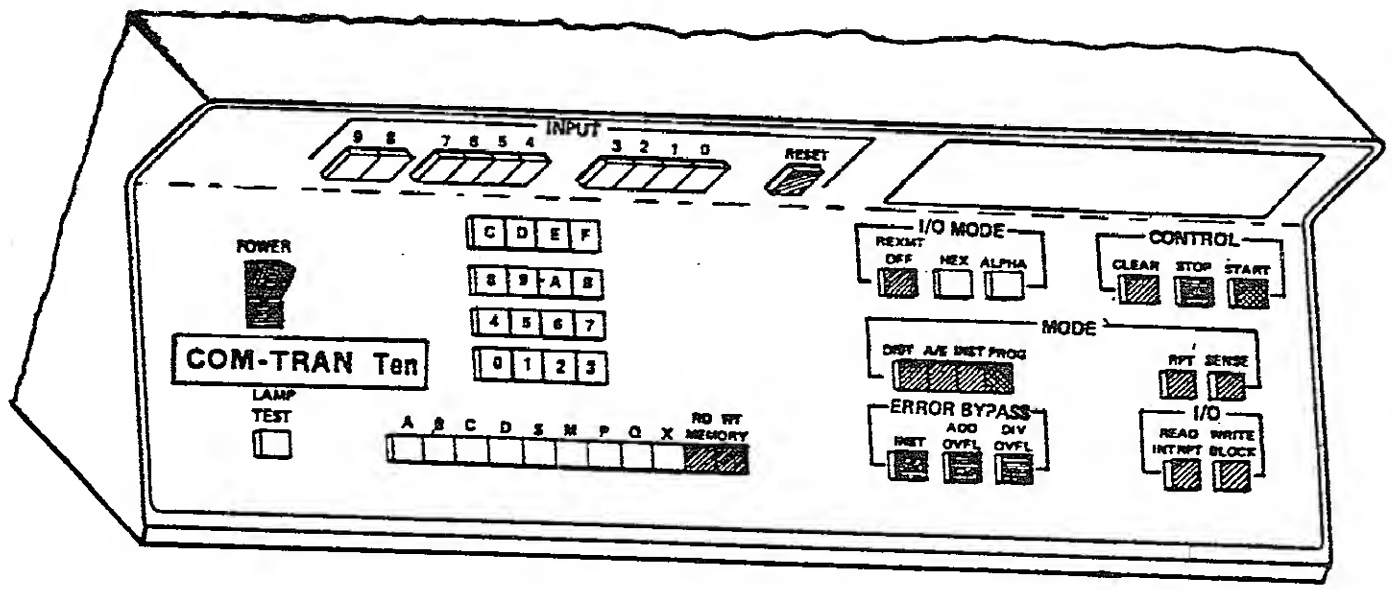


FIGURE 1 - COM-TRAN Ten control panel.

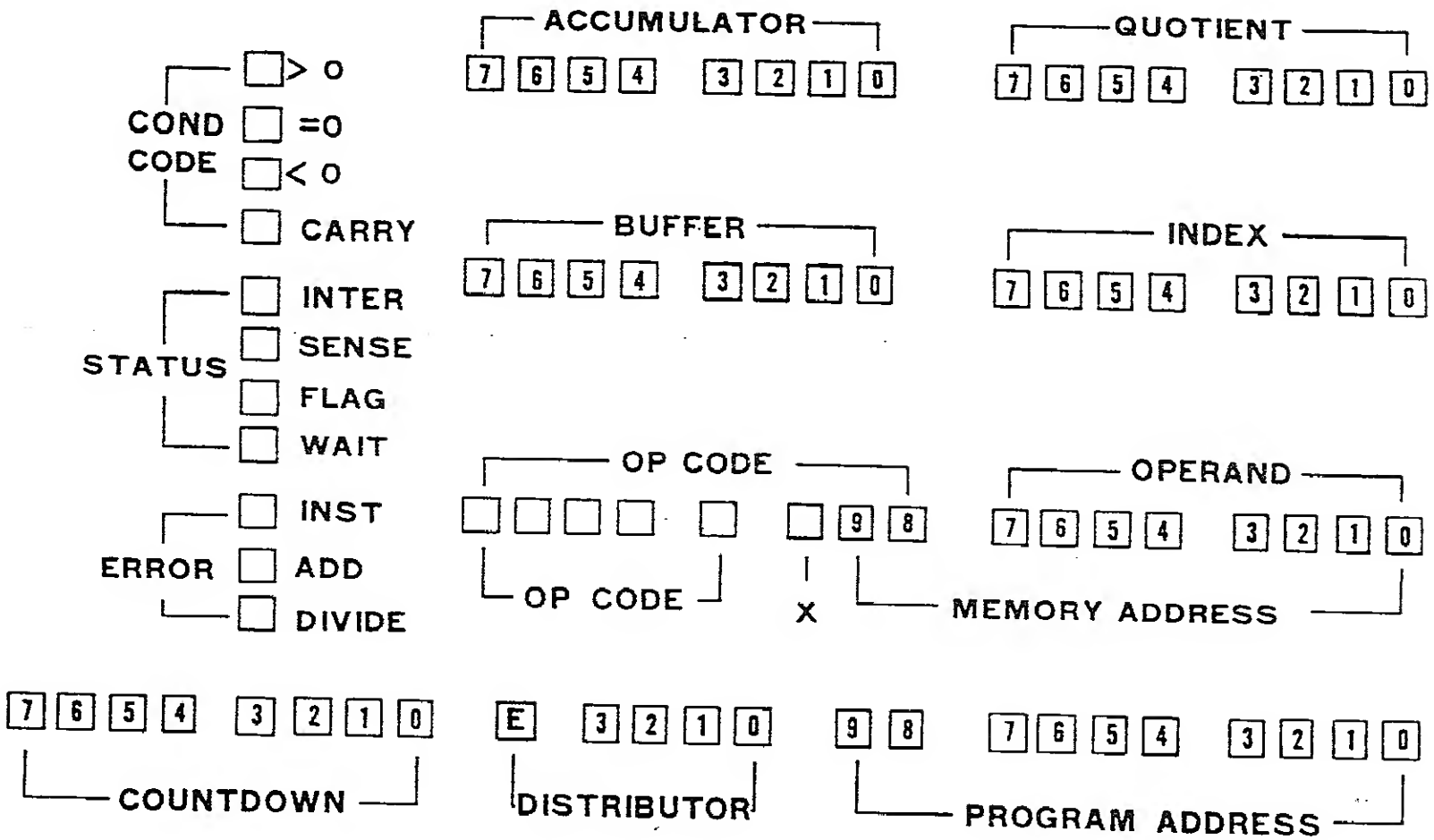


FIGURE 2 - Com-Tran 10 Display Panel.

III. Display Panel

A. Registers

B. Condition code indicators

C. Status indicators

D. Error detection indicators

IV. Instruction Repertoire

V. Logic Timing

NOTETAKING SHEET 5.5.1N

COM-TRAN Ten LOGIC AND DATA FLOW

REFERENCE:

Digiac Corporation. COM-TRAN Ten Technical Operations Manual M104, Vol. I, 1972. Section IV, pages 38-136.

NOTETAKING OUTLINE

I. Key to Block Diagram

II. Signal Busing

III. Input Switches

IV. I-register (input)

V. B-register (buffer)

VI. The 2's Complementor

VII. P-register (program address)

VIII. S-register (op-code)

IX. Instruction Decoder (control "I" circuits)

X. M-register (memory address)

XI. Memory Unit

XII. X-register (index)

XIII. Index Adder

XIV. Arithmetic Logic Unit (ALU)

XV. A-register (accumulator)

XVI. Q-register (quotient)

XVII. C-register (countdown)

XVIII. D-register (distributor)

XIX. Control Logic Circuits

XX. Clock (500 kHz)

NOTETAKING SHEET 5.6.1N

COM-TRAN Ten SOFTWARE

REFERENCES:

1. Digiac Corporation. Programming Manual for the COM-TRAN Ten, M-107, 1972. Section III, pp 24-36; section IV, pp 37-77.

2. Digiac Corporation. Technical Operations Manual for the COM-TRAN Ten, M-104, Volume I, 1972. Section III, pp 23-31; section IV, pp 64-127, 171-186.

3. Digital Computer Basics. NAVEDTRA 10088B. 1978. Chapter 9, pp. 210-228.

NOTETAKING OUTLINE

I. Programming Concepts

A. Computer program

B. Steps in programming

1. Statement

2. Analysis

3. Flow chart

4. Coding

5. Debugging

6. Documentation

II. Common Flowchart Symbols

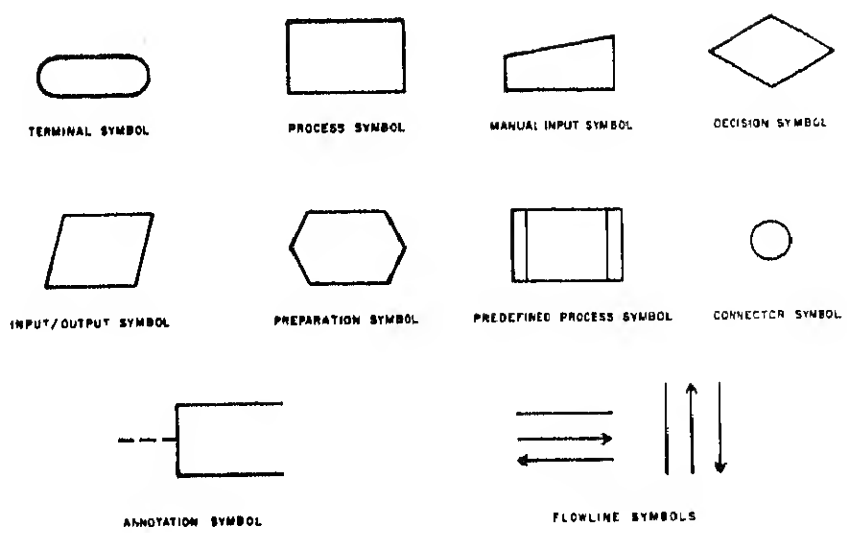


FIGURE 1 - Common flow chart symbols

A. Terminal symbol

B. Process symbol

C. Manaual input symbol

D. Decision symbol

E. Input/output symbol

F. Preparation symbol

G. Annotation symbol

H. Predefined process symbol

I. Connector symbol

J. Flow line symbols

III. Program Coding Conventions

A. Instruction code representation

INSTRUCTION NAME	MNEMONIC CODE	NUMERIC CODE	BINARY CODE
ADDition	ADD	$6D_{(16)}$	0110 0000
Shift Right Arithmetic	SRA	$1C_{(16)}$	0001 0000
LoaD Accumulator	LDA	$E0_{(16)}$	0010 0000
Branch and STop	BST	$98_{(16)}$	1001 1000

FIGURE 2 - Representation, mnemonic, numeric, and binary.

1. Mnemonic codes

2. Numeric code

3. Binary code

B. Instruction word format

1. OP CODE

2. OPERAND

C. Immediate instructions

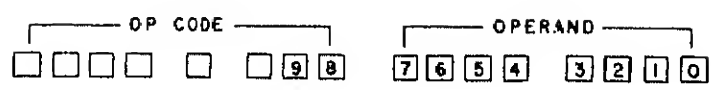
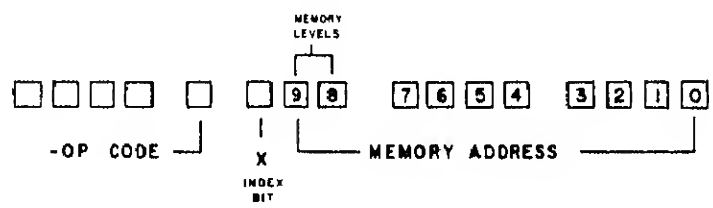


FIGURE 3A - Immediate instruction format.

D. Direct-indexed instructions



ct-indexed instruction format.

E. Addressing upper memory

HIGH ORDER BITS MEMORY ADDRESS		NUMBER ADDED TO OP-CODE		MEMORY LOCATIONS		
M9	M8					
0	0	0	FIRST LEVEL	000_{16}	TO	$0FF_{16}$
0	1	1	SECOND LEVEL	100_{16}	TO	$1FF_{16}$
1	0	2	THIRD LEVEL	200_{16}	TO	$2FF_{16}$
1	1	3	FOURTH LEVEL	300_{16}	TO	$3FF_{16}$

FIGURE 4 - Memory levels.

OP-CODE		OPERAND
20_{16}	FIRST LEVEL	AC_{16}
21_{16}	SECOND LEVEL	AC_{16}
22_{16}	THIRD LEVEL	AC_{16}
23_{16}	FOURTH LEVEL	AC_{16}

FIGURE 5 - Addressing memory levels for LDA instruction.

F. Index addressing

OP-CODE	OPERAND	INDEX REGISTER
24_{16}	30_{16}	00_{16}
24_{16}	30_{16}	01_{16}
24_{16}	30_{16}	03_{16}
24_{16}	30_{16}	04_{16}

FIGURE 6 - Index addressing for LDA instruction.

IV. Instruction Repertoire

A. General information

B. Types of instructions

1. Load instructions

2. Store instructions

3. Arithmetic instructions

4. Logical instructions

5. Branch instructions

6. Input/output instructions

V. Program Writing and Analysis

A. Basic programming

1. Flow chart

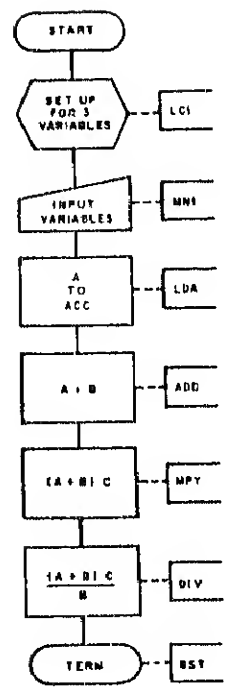


FIGURE 7A - Program flow chart.

2. Coding

COMPUTR PROGRAM FORM	PROGRAMMER:		PROGRAM TITLE: $\frac{(A+B)C}{B}$	
	DATE		PAGE:	NOTES:

PROGRAM ADDRESS	OPERATION CODE		OPERAND		REMARKS
	MNEMONIC	"HEX" CODE	MEMORY ADDRESS M	DATA WORD K	
000	LCI	01		02	LOAD C WITH ONE LESS THEN NUMBER OF VARIABLES
002	MNI	FD	0E		MANUAL INPUT VARIABLES STARTING AT ADDRESS 00E
004	LDA	20	0E		LOAD ACCU WITH CONTENTS OF ADDRESS 00E
006	ADD	60	0F		ADD A + B LEAVE SUM IN ACCU.
008	MPY	70	10		MPY A+B BY CONTENTS OF ADDRESS 010
00A	DIV	78	0F		DIV (A+B) C BY CONTENTS OF ADDRESS 00F
00C	BST	98	00		BRANCH AND STOP AT ADDRESS 000
00E		A	B		VARIABLES STORED
010		C			

FIGURE 7B - $\frac{(A+B)C}{B}$ Machine-coded.

B. Advanced programming

1. Flow chart

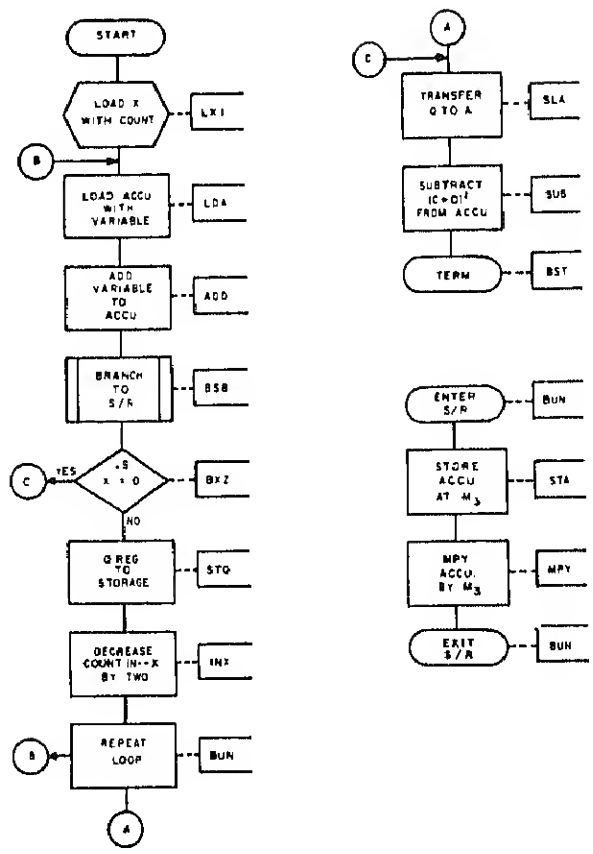


FIGURE 8A - $(A+B)^2 - (C+D)^2$
Program flow chart and subroutine.

2. Coding

COMPUTER PROGRAM FORM | PROGRAMMER: | DATE: | PAGE: | PROGRAM TITLE: $(A+B)^2-(C+D)^2$ | NOTES:

PROGRAM ADDRESS	OPERATION CODE MNEMONIC	OPERATION CODE "HEX" CODE	OPERAND MEMORY ADDRESS M	OPERAND DATA WORD K	REMARKS
100	LXI	12		02	LOAD X WITH COUNT OF TWO
102	LDA	27	01		LOAD A WITH CONTENTS OF ADDRESS 303 (INDEXED BY 2)
104	ADD	67	02		ADD CONTENTS OF ADDRESS 304 TO ACCU (INDEXED BY 2)
106	BSB	A2	00		BRANCH TO SQUARING SUBROUTINE
108	BXZ	C9	10		X=0 BRANCH TO ADDRESS 110
10A	STQ	58	00		STORE $(C+D)^2$ AT ADDRESS 300
10C	INX	03		FE	DECREASE X REG BY 2 (FE WILL SUBTRACT 2 FROM X REG.)
10E	BUN	91	02		REPEAT LOOP
110	SLA	08		08	SHIFT Q REG. IN TO ACCU.
112	SUB	69	00		SUBTRACT $(C+D)^2$ FROM ACCU
114	BST	99	00		RETURN TO START AND STOP
200	BUN	91	08		RETURN TO PROGRAM ADDRESS 108
202	STA	48	05		STORE ACCU. AT ADDRESS 305
204	MPY	73	05		SQUARE ACCU
206	BUN	92	00		RETURN TO START OF SUBROUTINE ADDRESS 200
300		$(C+D)^2$	A		VARIABLES AND TEMPORARY STORAGE
302		B	C		
304		D	M_3		

FIGURE 8B - $(A+B)2 - (C+D)^2$ Machine-coded.

NOTETAKING SHEET 5.7.1N

COM-TRAN Ten HARDWARE AND LOGIC DIAGRAM DATA FLOW

REFERENCES:

Technical Operations Manual for the COM-TRAN Ten, M-104, Volume I. Digiac Corporation, 1972. Section IV, pp 59-136.

NOTETAKING OUTLINE

I. Key to logic Diagrams

A. Inputs

B. Outputs

C. Logic symbols

D. Phantom "OR" symbol

II. Input Section

A. Switch 1 circuits

B. Switch 2 circuits

C. Switch logic

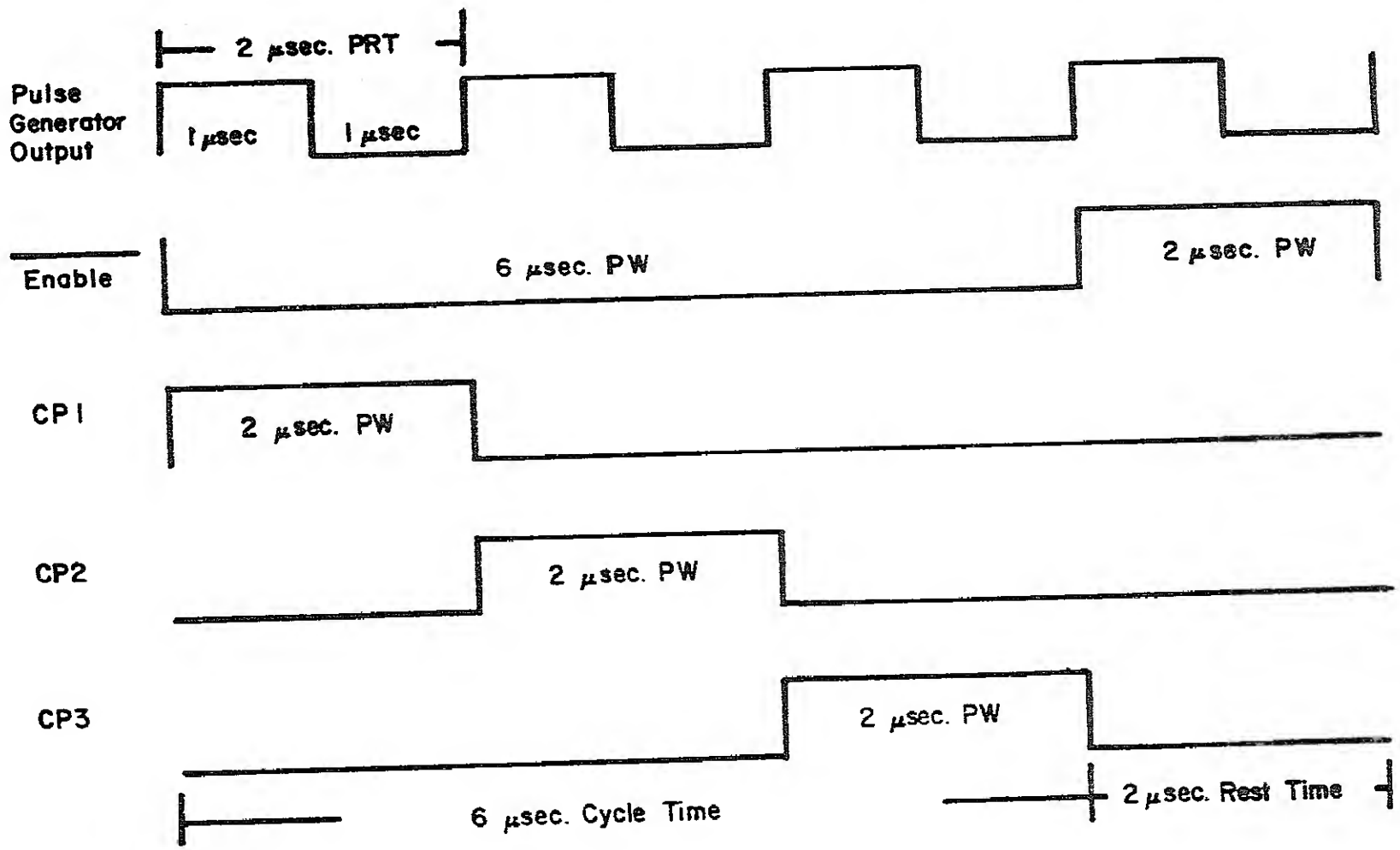
D. I-register

III. Control Section

A.

B. D-register

C. Decoder



Clock Timing Sequence

FIGURE 1

D. S-register

E. Instruction decoder

F. P-register

G. C-register

IV. Memory Section

A. M-register

B. Memory module

C. X-Register/x-Adder

00	SST,k	Sense status 8-bit status word → ACC
01	LCI,k	Load COUNTDOWN immediate k → (C)
02	LAI,k	Load ACCUMULATOR immediate k → (ACC)
03	INX,k	Increase INDEX (INDEX) + k →(INDEX)
08	SKI,k	Skip on INTERrupt k instructions
09	SKS,k	Skip on SENSE k instructions
0A	SKF,k	Skip on FLAG k instructions
0B	SLA,k	Shift LEFT Arithmetic (AQ) to left k places
10	SRA,k	Shift RIGHT Arithmetic (AQ) to right k places
11	OCD,k	Output Command k → external device
12	LXI,k	Load INDEX immediate k → (INDEX)
13	SLL,k	Shift LEFT Logical (ACC) to left k places
18	SRL,k	Shift RIGHT Logical (ACC) to right k places
19	AND,k	AND k AND (ACC) → (ACC)
1A	IOR,k	Inclusive OR k OR (ACC) → (ACC)
1B	XOR,k	Exclusive OR k EOR (ACC)→ (ACC)
20	LDA,m,x	Load ACCUMULATOR (m) → (ACC)
28	FLC,k	FLAG clear 0 → FLAG bit
30	LCC,m,x	Load Consecutive (m)→ (m+1)
38	LAN,m,x	Load ACCUMULATOR Negative -(m)→ (ACC)
40	LDQ,m,x	Load QUOTIENT register (m)→ (QUO)
48	STA,m,x	Store ACCUMULATOR (ACC)→ (m)
50	STX,m,x	Store INDEX (INDEX)→ (m)
58	STQ,m,x	Store QUOTIENT (QUO) → (m)
60	ADD,m,x	ADD (ACC) + (m)→ (ACC)
68	SUB,m,x	SUBtract (ACC) - (m) → (ACC)
70	MPY,m,x	Multiply (ACC) * (m) → (AQ)
78	DIV,m,x	DIVide (AQ) ÷ (m) → (Q) remainder → (ACC)
80	RAO,m,x	Replace Add One (m) + 1→ (m)
88	RSO,m,x	Replace Subtract One (m) - 1→ (m)
90	BUN,m,x	Branch UNconditional m → (PA)
98	BST,m,x	Branch & Stop m→ PA & STOP
A0	BSB,m,x	Branch to Subroutine 90 (BUN) → (m) (PA) → (m+1) (m+2)→ (PA)
A8	BPS,m,x	Branch on Positive > 0 → m → (PA)
B0	BZE,m,x	Branch on Zero = 0 → m → (PA)
B8	BNG,m,x	Branch on Negative <0 → m → (PA)
C0	BNC,m,x	Branch on No Carry CARRY = 0 → m → (PA)
C8	BXZ,m,x	Branch on INDEX Zero (INDEX) = 0→ m → (PA)
D0	WDB,m,x	Write Data Block (m) → external device
D8	MNO,m,x	Manual Output (m)→ (INPUT) & (BUFFER)
E0	RDB,m,x	Read Data Block Data→ (m)
E8	RDI,m,x	Read until Interrupt Data → (m)
F0	MNI,m,x	Manual Input (INDEX) & (BUFFER) → (m)
F8	FLS,k	FLAG Set 1→ FLAG bit

FIGURE 2 - Numerical listing of instructions.

V. Arithmetic Section

A. B-register/2's complemented/selector

B. A-register

C. ALU

D. Q-register

VI. Output Section

A. Lamp modules 1 and 2

B. Inputs to lamp module 1

C. Outputs from lamp module 1

D. Inputs to lamp module 2

E. Outputs from lamp module 2

VII. Logic Timing

JOB SHEET 5.8.1J

COM-TRAN Ten DATA FLOW ANALYSIS

INTRODUCTION:

The purpose of this job sheet is to reinforce the classroom instruction on the COM-TRAN Ten and to familiarize you with its operation while the program is being monitored step by step through both the acquisition and execution phase of an instruction.

LESSON TOPIC LEARNING OBJECTIVES:

11.4.1. SELECT, from a given list, the statement describing the actions of the COM-TRAN Ten during the acquisition phase.

11.4.2. SELECT, from a given list, the statement describing the actions of the COM-TRAN Ten during the execution phase.

11.4.3. ANALYZE, a program using COM-TRAN Ten instructions, by listing the contents of the registers at any specified time during or after the execution of acquisition phase of the program.

11.4.4. SELECT, from a given list with the aid of the HSI and the COM-TRAN Ten program, the statement correctly describing COM-TRAN Ten operations and data flow.

REFERENCES:

1. Digiac Corporation. Programming Manual for the COM-TRAN Ten, M-107. 1972. Section II, pp 29-36; section IV, pp 37-77.

2. Digiac Corporation. Technical Operations Manual for the COM-TRAN Ten, M104, Volume I, 1972. Section III, pp 34-35; Section IV, pp 52-127.

EQUIPMENT AND MATERIALS:

COM-TRAN Ten Digital Computer Training Device (6F21).

JOB STEPS:

Precautions to be observed

1. Remove all exposed jewelry
2. No food or drink allowed in the lab area.
3. Instructor MUST be present when the equipment is energized.

Step 1. Initial equipment setup

a. To obtain this condition, refer to paragraph 3-6 of the HSI.

b. Complete the "Daily Inspection Check" as listed before continuing.

Step 2. Manual loading

a. To load the program manually, refer to paragraph 3-2-1 of the HSI, table VII.

b. Load the data listed in the table below in the computer.

Program Address	Mnemonic Code	OP CODE	OPERAND
100	LAI	02	07
102	LXI	12	01
104	MPY	71	08
106	DIV	7D	07
108		03	

Step 3. Executing the Program

a. Acquisition of the Load Accumulator Immediate (LAI) instruction

(1) Refer to paragraph 3-3 of the HSI, table IX.

(2) Use part 3 of the table and complete this portion in the A/E mode.

(3) Record the contents of the following registers;

S-register _____ _____

M-register _____ _____

P-register ____ _____ _____

D-register ____ _____

b. Execution of the (LAI) instruction

(1) Depress START switch. Ensure instruction was properly executed.

(2) Record the contents of the following registers;

A-register _____ _____

D-register _____ _____

c. Acquisition of the Load Index Register Immediate (LXI) instruction

(1) Depress START switch. Ensure that the proper OP CODE and OPERAND are displayed on the panel.

(2) Record the contents of the following registers;

S-register _____ _____

M-register _____ _____

P-register _____ _____ _____

D-register _____ _____

d. Execution of the LXI instruction

(1) Depress START switch. Ensure that the instruction was properly executed.

(2) Record the contents of the following registers;

X-register _____ _____

D-register _____ _____

e. Acquisition of the Multiply (MPY) instruction

(1) Depress the START switch. Ensure that the proper OP CODE and OPERAND are displayed on the panel.

(2) Record the contents of the following registers:

S-register _____ _____

M-register _____ _____ _____

P-register _____ _____ _____

D-register _____ _____

Instructor's Initials _____

f. Execution of the MPY instruction by individual distributor pulses. Refer to paragraph 4-3-14 of the HSI for Logic Timing.

(1) Select DIST mode. The computer is now ready to execute the multiply program.

(2) DPO

(a) Depress START.

(b) Record the contents of the following registers:

B-register _____ _____

C-register _____ _____

D-register _____ _____

(3) DP1 (Note: IF B_7 = 0, computer skips DP2 and DP3)

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(4) DP4

(a) Depress START.

(b) Record the contents of the following registers:

B-register _____ _____

D-register _____ _____

(5) DP5

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(6) DP6

(a) Depress START.

(b) Record the contents of the following:

Q0 bit _____

D-register _____ _____

(7) DP7

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

B-register _____ _____

D-register _____ _____

(8) DP8

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(9) DP9

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

Q-register _____ _____

D-register _____ _____

(10) DP10

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(11) DP11

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(12) DP6

(a) Depress START.

(b) Record the contents of the following:

Q_0 bit _____

D-register _____ _____

(13) DP7

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

Q-register _____ _____

D-register _____ _____

(14) DP8

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(15) DP9

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

Q-register _____ _____

D-register _____ _____

(16) DP10

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(17) DP11

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(18) DP6

(a) Depress START.

(b) Record the contents of the following:

Q0 bit _____

D-register _____ _____

(19) DP9

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

Q-register _____ _____

D-register _____ _____

(20) DP10

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(21) DP11

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(22) DP6

(a) Depress START.

(b) Record the contents of the following:

Q_0 bit _____

D-register _____ _____

(23) DP9

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

Q-register _____ _____

D-register _____ _____

(24) DP10

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(25) DP11

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(26) DP6

(a) Depress START.

(b) Record the contents of the following:

Q_0 bit _____

D-register _____ _____

(27) DP9

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

Q-register _____ _____

D-register _____ _____

(28) DP10

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

(29) DP11

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(40) DP10

(a) Depress START.

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(41) DP11

(a) Depress START

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(42) DP12
DP13 (No operation)
DP14

(43) Continue depressing START until DP15 is displayed in the D-register (1F16).

(44) DP15

(a) Depress START.

(b) Record the contents of the following:

Condition code _____

D-register _____ _____

A- and Q-registers _____ _____; _____

_____

Instructors initials _____

g. Acquisition of the Divide (DIV) instruction by distributor pulses. (Refer to paragraph 4-2-1 of the HSI for logic timing.)

(1) Leave in the DIST Mode. The computer is now ready to acquire the DIV instruction.

(2) Three facts regarding acquisition;

(a) Always precedes execution.

(b) Is the same for all instructions.

(c) Is the phase where indexing takes place.

(3) DPA0

(a) Depress START.

(b) Record the contents of the following registers:

M-register _____ _____ _____

D-register _____ _____ _____

(4) DPA1

(a) Depress START.

(b) Record the contents of the following registers:

B-register _____ _____

D-register _____ _____

(5) DPA2

(a) Depress START.

(b) Record the contents of the following registers:

S-register _____ _____

D-register _____ _____

(6) DPA3

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(7) DPA4

(a) Depress START.

(b) Record the contents of the following registers:

M-register _____ _____ _____

D-register _____ _____

(8) DPA5

(a) Depress START.

(b) Record the contents of the following registers:

B-register _____ _____

D-register _____ _____

(9) DPA6

(a) Depress START.

(b) Record the contents of the following registers:

M-register _____ _____ _____

D-register _____ _____

(10) DPA7

(a) Depress START.

(b) Record the contents of the following registers:

P-register _____ _____ _____

D-register _____ _____

(11) DPA8

(a) Depress START.

(b) Record the contents of the following registers:

M-register _____ _____ _____

D-register _____ _____

(12) DPA9

(a) Depress START.

(b) Record the contents of the following registers:

M-register _____ _____ _____

D-register _____ _____

(13) DPA10 (NOTE: DPA 11 through DPA 14 have no operation.)

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(14) DPA15

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

S-register _____ _____

M-register _____ _____ _____

NOTE: Have instructor verify procedure. Date: _____

Instructor initials: _____

h. Execution of the Divide (DIV) instruction by distributor pulses. (Refer to paragraph 4-3-15 of the HSI for logic timing.)

(1) Leave in the DIST mode. The computer is now ready to execute the DIV instruction.

(2) Division is performed by the shift and add/subtract method.

(3) DP0

(a) Depress START.

(b) Record the contents of the following registers:

B-register _____ _____

C-register _____ _____

D-register _____ _____

(4) DP1

(a) Depress START.

(b) Record the contents of the following registers:

D-register _____ _____

(5) DP2 (NOTE: If $A_7 = 0$, computer skips to DP6.)

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(6) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(7) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(8) DP8

(a) Depress START.

(b) Record the contents of the following register:

A-register _____ _____

D-register _____ _____

(9) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(10) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(11) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(12) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(13) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(14) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(15) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(16) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(17) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(18) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(19) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(20) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(21) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(22) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(23) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(24) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(25) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(26) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(27) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(28) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(29) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(30) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(31) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(32) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(33) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(34) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(35) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(36) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(37) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(38) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(39) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(40) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(41) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(42) DP7

(a) Depress START.

(b) Record the contents of the following registers:

C-register _____ _____

D-register _____ _____

(43) DP8

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(44) DP9

(a) Depress START.

(b) Record the contents of the following registers:

Q-register _____ _____

D-register _____ _____

(45) DP10

(a) Depress START.

(b) Record the contents of the following registers:

A-register _____ _____

D-register _____ _____

(46) DP6

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(47) DP11

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(48) DP12

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(49) DP13

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(50) DP14

(a) Depress START.

(b) Record the contents of the following register:

D-register _____ _____

(51) DP15

(a) Depress START.

(b) Record the contents of the following:

Condition code _____

D-register _____ _____

A- and Q-registers _____ _____, _____

_____

NOTE: Have instructor verify procedure. Date: _____

Instructor initials: _____

SELF-TEST ITEMS

1. The purpose of the acquisition phase is to acquire the _____ and the _____ for execution.

2. The bit used to determine Initiate Add (INA occurs during the MPY instruction and is the Q_0/F_7 bit). (Choose one.)

3. During DP1 of the MPY instruction the data are tested for polarity. If found negative, the 2's complementor would/would not be used. _____ (Choose one.)

4. The "condition codes" are set at the end of the _____ phase.

5. Which pulse of the acquisition phase causes the contents of the X-register to be added to the contents of the M-register?
   _____

6. During MPY, SC8 is generated by _____ and _____.

7. DP15 of MPY occurs only when _____ = _____.

8. Which bit controls whether the COM-TRAN Ten adds or subtracts during the DIV operation? _____

9. With the exception of the shifting required to attain proper position, at which step on the Job Sheet of the MPY instruction was the correct answer formed? _____

10. "Indexing" is accomplished by adding the contents of the ___-register to the contents of the _____-register during during DPA _____.

11. DPA12 will be displayed in the D-register during the acquisition.
    ____True. ____False.

12. During the Execution phase of DIV, if the computer skips from DP2 to DP6, we know the _____ is positive.

13. The COM-TRAN Ten used both addition and subtraction to execute this DIV funciton? _____True. _____False.

14. Which distributor pulses make up the Divide Group (DVG)?

_____.

15. The "remainder" is the 2's complement when the _____ is negative.

16. A DIV error signal may be generated at DP_____, DP_____, or DP_____.

17. List both the MPY and DIV problems in decimal form.

_____ x _____ = _____; _____ ' _____ =

_____

18. During the DIV process, when will a 1 be shifted into the Q-register?

When _____ = _____.

NOTETAKING SHEET 5.8.1N

COM-TRAN Ten DATA FLOW ANALYSIS

REFERENCES:

1. Digiac Corporation. Programming Manual for the COM-TRAN Ten, M-107, 1972. Section II, pp 29-36; section IV, pp 37-77.

2. Digiac Corporation. Technical Operations Manual for the COM-TRAN Ten, M-104, Volume I. 1972. Section III, pp 34-35; section IV, pp 52-127.

NOTETAKING OUTLINE

I. Safety

A. Discussion

B. Lab Safety

1. Personal safety items

2. Equipment safety items

II. Job Sheet 5.8.1J

A. Three stages

1. Initial equipment setup

2. Manual loading

3. Executing the program

B. Three processes

1. Acquisition phase

2. Execution of multiply

3. Execution of divide

C. Additional requirements

III. Stand-by and Secure Procedures

A. The stand-by condition

B. The secure condition

C. Usage

NOTETAKING SHEET 5.9.1N

COM-TRAN Ten FAULT ISOLATION

REFERENCES:

1. Digiac Corporation. COM-TRAN Ten Technical Operations Manual M104, Vol. I, 1972.

2. Digiac Corportion. Programming Manual for the COM-TRAN Ten, M104. 1972.

LESSON TOPIC LEARNING OBJECTIVES

11.2.1 PERFORM a daily inspection on a digital computer trainer, using a given HSI as reference.

11.3.40 PERFORM a preoperational check on a digital computer trainer, using a given HSI as reference.

11.3.41 PERFORM an operational check on a digital computer trainer, using a given HSI as reference.

11.4.5 ISOLATE, with the use of given test equipment, logic diagrams, and an HSI, an instructor-induced malfunction, following all procedures outlined in the HSI for the digital computer trainer 6F21 and observing all safety precutions.

11.4.6 DOCUMENT, on the data sheet provided, using given test equipment and an HSI: abnormal indications, failing subroutine, failing instruction, failing phase of instruction, failing distributor phase, failing subcommand, and problem location.

11.5.1 DOCUMENT, on the VIDS/MAF provided, all necessary corrective actions to simulate restoration of an instructor-induced malfunction on a digital computer trainer.

NOTETAKING OUTLINE

I. Safety

A. Personal safety items

B. Equipment safety

II. Test Equipment Setup

A. VIDS/MAF and data sheet entries

B. Logic probe lamp indications

C. Troubleshooting Procedures

1. Step 1: Daily inspection

2. Step 2: Preoperational check

3. Step 3: Operational check

4. Step 4: Subroutine check

5. Step 5: Acquisition/execution check

6. Step 6: Isolation to problem

7. Step 7: Signal-tracing

III. VIDS/MAF

IV. Standby and Secure Procedures

A. Standby procedure

B. Secure procedures